RetroMagazine
Anno 2 - Numero 6

APRILE 2018 - WWW.RETROMAGAZINE.NET

IN EVIDENZA IN QUESTO NUMERO

THEC64 Mini: il pomo della discordia!

di Francesco Fiorentini

Chiudete gli occhi ed immaginate di essere di nuovo quel giorno di tanti anni fa in cui riceveste in regalo il vostro Commodore 64. Ripensate al momento in cui per la prima volta lo avete collegato al televisore di casa ed e' comparsa la schermata blu che tutti noi oggi conosciamo molto bene. Subito dopo avete inserito la cassetta del vostro primo gioco nel datassette e dopo un tempo di attesa che adesso ci sembrerebbe eterno, un'esplosione di suoni e colori (per i fortunati possessori di un televisore a colori) ha invaso la stanza accendendo definitivamente una passione che non si sarebbe mai piu' sopita!

Quanti di noi si riconoscono in questa storia? Quanti di noi hanno mosso i primi passi videoludici con l'amato biscottone? Tanti, forse la maggioranza dei lettori di RetroMagazine... Ebbene e' proprio sull'onda di questo effetto nostalgia che la Retro Games LTD ha lanciato l'operazione THEC64 Mini che tanto ha tenuto banco ultimamente nelle discussioni dei piu' frequentati gruppi Facebook dedicati al Retrocomputing ed al Retrogaming.

Ovviamente c'e' chi ha visto in questa riproposizione un attentato di lesa maesta' verso un mito indiscusso e chi invece ha accolto favorevolmente questa operazione commerciale. Perche' in fin dei conti, sempre di un'operazione commerciale si tratta. Non sta certo a me giudicare chi ha torto e chi ha ragione, pero' per fortuna posso esprimere il mio pensiero.

Devo ammettere che all'inizio ero scettico riguardo al prodotto in questione, soprattutto per le ridotte dimensioni e la conseguente mancanza della tastiera; come se si volesse ridurre un Home Computer ad una console, cancellando in un colpo solo le notti insonni che tanti di noi hanno passato di fronte al C64 a scrivere programmi in BASIC o in Assembly. Pero', come mi faceva notare **Marco Pistorio**, in ben pochi adesso scrivono codice sul C64 reale, preferendo tool piu' evoluti come i cross compilatori. E forse e' una delle ragioni che ha spinto la Retro Games LTD a puntare soprattutto sul lato ludico del C64. E da questo punto di vista il THEC64 Mini e' indubbiamente pratico per il retrogamer neofita o per il millenial che si avvicina al C64 per la prima volta. E' ovvio che il target principale del prodotto non puo' essere lo smanettone appassionato di retrocomputing; ma quanti di noi alla fine resisteranno alla tentazione di comprarlo anche solo per avere un oggetto che ricorda nella forma un mito? Io no, lo confesso! Ancora dubbi? Andate a leggere la recensione a pagina 16. ☺

Amiga - Il Chipset

Dante Profeta, grande appassionato di Amiga, torna a parlare del 16 bit piu' iconico in assoluto e lo fa alla grande, introducendoci alle sue modalita' grafiche.

Articolo a pagina 9

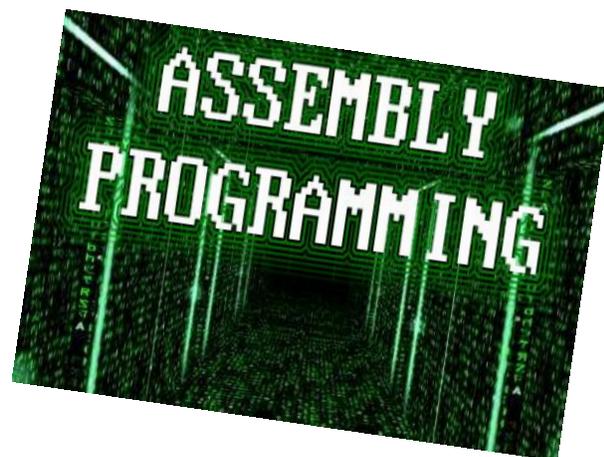
Intervista doppia

Il nostro David La Monaca (aka Cercamon) si e' messo in contatto con Bonaventura Di Bello e Marco Vallarino, scrittori e autori di avventure testuali e della nuovissima fiction interattiva intitolata "Déjà Vu" e ne e' nata un'intervista doppia stile "Le Iene" che siamo sicuri apprezzerete!

Articolo a pagina 24

Una intro... per iniziare!

Programmazione Assembly su Commodore 64

di Marco Pistorio

Parte I – Introduzione

Le demo nacquero, in origine, come firme introdotte nei programmi o nei supporti di installazione di programmi e giochi da parte di alcuni cracker, i cosiddetti pirati informatici, capaci di scardinare le difese elettroniche poste a guardia del software.

In un'epoca in cui pochi paesi al mondo avevano una legislazione che proteggeva il software dalla copia, eliminare le protezioni e diffondere un programma era un'operazione comune.
Chiunque avesse un minimo di capacità informatica copiava i programmi, ma soltanto coloro che avevano maggiore abilità, inventiva e pazienza riuscivano a introdurre la propria demoscene personalizzata nelle proprie copie: questo per distinguersi dalla massa, facendosi conoscere e facendosi riconoscere nelle proprie copie illegali diffuse.

Queste firme, spesso colorate ed arricchite con una base musicale accattivante, erano realizzate in linguaggio macchina e memorizzate all'interno dello stesso supporto contenente il programma copiato, a volte "fondendosi" con quest'ultimo. Il particolare tipo di animazioni che partiva al lancio del programma copiato era denominato "letter", "sign", "message" o "**intro**".

Lo scopo di questo articolo è quello di presentare una semplice intro, illlustrando tutto il codice assembly necessario, codice che cercherò di spiegare nella maniera più semplice possibile.
L'articolo sarà suddiviso in "puntate", per evitare di realizzare un unico articolo particolarmente lungo e difficile da leggere.

La intro che Vi presenterò non sarà completa al 100% perché il mio obiettivo è presentarVi una intro con un codice più breve e semplice possibile.
Non ho previsto quindi alcun testo a scorrimento orizzontale.
La intro conterrà due rasterbars in movimento, elemento presente nella maggioranza delle intro. Tuttavia le rasterbars risulteranno leggermente sfrangiate, non perfettamente orizzontali.
Ciò perché nel codice ho adottato una semplice tecnica per ottenere tali rasterbars. Maggiore semplicità al costo però di una precisione approssimativa.
La intro che Vi presenterò quindi non riscuoterà particolare successo nei confronti di tutti coloro che conoscono già queste tematiche in maniera più approfondita e che realizzano per contro proprio intro, giochi etc.
Tuttavia mi auguro che altri la apprezzino. Chi, per esempio, ha visto diverse intro ma non ne ha mai realizzato nessuna, e che potrà quindi sfruttare la lettura di queste pagine di "RetroMagazine" per realizzarne una.

L'argomento trattato è particolarmente tecnico. Nonostante i miei sforzi, alcuni passaggi potrebbero risultarVi non chiari. Una infarinatura di base relativa all'assembly Vi sarà di notevole aiuto nell'affrontare tutti i passaggi un po' più complessi che affronteremo.
Oggi, a differenza di quanto accadeva negli anni '80, è facile ed immediato ottenere tutte le informazioni di cui necessitiamo, basta effettuare ricerche "mirate" in rete. Potrete ottenere comunque tutti i ragguagli che potremo fornirVi scrivendo, come sempre, alla nostra email redazionale.

Parte II – Elementi di Assembly

Come scrivevo in precedenza, una infarinatura di base relativa alla programmazione assembly Vi sarà di aiuto nel seguire meglio la spiegazione del codice relativo alla intro.
Tuttavia, a beneficio soprattutto di coloro che non abbiano ancora aquisito tali conoscenze, cercherò di fornirVi alcuni elementi di programmazione assembly, cercando di esporre il tutto nella maniera più semplice possibile, senza alcuna intenzione di presentare un corso serio e rigoroso al riguardo.

Iniziamo questa "chiacchierata" tra amici domandandoci il perché sia opportuno scrivere un programma in assembly piuttosto che, ad esempio, in BASIC, per realizzare una intro.
Il motivo è sostanzialmente uno. Grazie all'assembly riusciamo a gestire in modo completo, ed alla massima velocità oltretutto, tutte le caratteristiche del C64. In più, volendo, potremmo anche fare a meno dell'interprete BASIC che potremmo disattivare liberando la RAM al di sotto dello stesso. Poca roba (8 Kbytes, precisamente) ma lavorando con il C64 un po' di RAM a disposizione in più può tornare sempre utile.
Pro dell'assembly? Potremmo dire accesso a tutte le caratteristiche del C64 con estrema velocità.
Contro dell'assembly? Essenzialmente la difficoltà di sviluppare in assembly e la poca 'portabilità' del codice ovvero un codice scritto per un C64 in assembly potrà girare perfettamente solo su C64. Su un VIC20, che monta un processore molto simile, in linea teorica il codice girerebbe allo stesso modo ma la mappa di memoria è diversa, è diverso il comparto audio/video quindi è probabile che il codice andrebbe rivisto, in parte oppure in toto.
Su una macchina ad 8 bit con diverso processore invece, come ad esempio il Sinclair ZX Spectrum, che monta un microprocessore Z80, il codice si dovrebbe necessariamente riscrivere ex-novo.

Registri: A (Accumulatore) e Registri X ed Y (di indirizzamento)

Cosa sono i registri? I registri contengono un dato, sempre numerico.
Immaginateli un po' come delle variabili numeriche del BASIC.
In particolare, il registro A è quello più importante, perché sul contenuto di questo registro è possibile effettuare operazioni logiche (AND, OR per esempio) ed aritmetiche (Addizione e Sottrazione).

I registri X ed Y permettono anch'essi di memorizzare un valore numerico, permettono anche l'incremento oppure il decremento di tale valore numerico memorizzato all'interno del registro ma niente operazioni logiche né aritmetiche.
Vengono adoperati per l'indirizzamento, concetto di cui parleremo meglio più avanti, osservando il codice della intro di esempio.

Registro di Stato del processore

Questo registro 'speciale' mostra lo stato legato al risultato di operazioni logico/matematiche. E' costituito da 8 flags, 8 bit che possono assumere il valore '0' oppure '1' in funzione del fatto che l'ultima operazione matematica su un registro ne abbia azzerato il suo valore, oppure lo abbia reso negativo, oppure ancora abbia fatto oltrepassare il valore massimo consentito all'interno del registro stesso, che è 255 nel caso di registri ad 8 bit.

La condizione di registro contenente il valore 0 è rispecchiata dal flag Z del Registro di Stato del processore. La condizione che invece è stato ottenuto un valore negativo è rispecchiata dal valore contenuto nel flag 'N'. Infine la condizione che è stato superato il valore massimo che è possibile contenere all'interno del registro è rispecchiata dal valore del flag 'C'.

Nel caso del microprocessore 6510, quello contenuto nel Commodore 64 per intenderci, degli 8 flags presenti nel registro di stato del processore ne vengono adoperati soltanto 7. Uno non ha alcun significato assegnato e resta sempre al valore logico di '1'.

Lo "stack"

Qualora ve ne sia la necessità, è possibile memorizzare il contenuto di un registro sullo stack per poi prelevarlo successivamente.
Lo stack è una memoria di appoggio di tipo "L.I.F.O." (Last In, First Out) ovvero l'ultimo dato inserito all'interno dello stack è il primo che sarà possibile tirar fuori dallo stesso.
Ricordate che, nel caso del microprocessore 6510 in particolare, è possibile memorizzare e riottenere, quando necessita nuovamente, solo il contenuto di questi registri : Accumulatore, Registro X e Registro di Stato del microprocessore. Non è possibile quindi usare TUTTI i registri insieme allo stack.

Finisce qui questa brevissima carrellata volta a presentarVi velocemente gli attori coinvolti nella programmazione Assembly del microprocessore 6510 in particolare.

Alcuni argomenti li tratteremo commentando il codice.
Vedremo alcuni metodi di indirizzamento, quelli più semplici che ho utilizzato all'intero del codice della intro, ma non tutti.
Ci soffermeremo su alcuni argomenti successivamente, nelle prossime "puntate" quando ad esempio otterremo l'effetto di scorrimento dei due muri verso sinistra e verso destra.
Per tutto il resto, non mettiamo limiti alla Provvidenza, continuate a seguirci, mi raccomando!

E' necessario però aggiungere due righe relativamente al "raster", argomento già trattato nello scorso numero di "RM", il numero 5.

Lo schermo video viene continuamente ridisegnato (con una frequenza di 50 Hz,ovvero 50 volte al secondo, oppure di 60 Hz, 60 volte al secondo, in funzione del sistema video in uso, PAL oppure NTSC) riga per riga, indipendentemente dal fatto che l'immagine riprodotta nel frattempo cambi o meno.
Tale operazione viene effettuata da un cannone laser che percorre il reticolo di linee che compone lo schermo video (*raster* è il termine inglese che sta appunto per trama, reticolo, griglia).
Il cannone laser eccita elettricamente il materiale di cui è composto lo schermo e vedremo quindi apparire, linea per linea, l'immagine sullo schermo.

I primi 8 bit che descrivono il numero di riga che sta disegnando il raster in tempo reale sono contenuti all'interno di una locazione di memoria ben precisa, la locazione 53266 ($d012) in notazione esadecimale). Esiste un nono bit, posto ad '1' solo quando il raster si trova a disegnare numeri di riga oltre la 256-ma che si trova all'interno della locazione 53265 ($d011) ma nella intro ignorerò questi casi.

Non vi tedio ulteriormente ma ricordo a tutti i lettori interessati che tramite ricerche "mirate" su internet sarà loro possibile trovare tutta la documentazione che riterranno utile per approfondire gli argomenti accennati.

Parte III – I *ferri* del mestiere

Dopo aver parlato dei vantaggi di programmare in assembly rispetto ad altri linguaggi ed aver fatto una veloce disamina del funzionamento dei registri del microprocessore e dello stack, vediamo adesso come scrivere il codice assembly.

Basta un **editor di testo** ed un **compilatore**.
Altri preferiscono adoperare appositi ambienti di sviluppo integrati (**I.D.E.**) che includono al loro interno l'editor di codice, il compilatore ed altri strumenti a disposizione del programmatore, per creare schermate, per realizzare set di caratteri ridefiniti etc.

Io personalmente adopero come editor il gedit di Linux, senza particolari ammenicoli, e KickAssembler come compilatore, al quale do in pasto il file di testo contenente il codice assembly che ho scritto. KickAssembler, se è tutto ok, mi genererà il file .prg che contiene il codice macchina da lanciare tramite VICE, ma che potrei eseguire anche su un C64 reale (previo suo trasferimento in RAM chiaramente).

Perchè un editor di testo e non un ambiente di sviluppo integrato (I.D.E.)?
E' una mia scelta personale, condivisibile o meno.
L'uso di un IDE sottintende che tutto il lavoro di sviluppo debba essere effettuato al suo interno. Ma a me piace adoperare tools specifici per compiti specifici, ad esempio uso normalmente il CharPad per lavorare con i caratteri ridefiniti, evitando di imparare ed utilizzare solo un certo IDE (con tutti i suoi eventuali strumenti integrati più o meno potenti) e basta.
Utilizzando un semplice editor di testo faccio a meno anche del controllo automatico della correttezza del codice mentre lo sto scrivendo (il lavoro che fa l'Intellisense, per chi di Voi conosce l'ambiente di sviluppo Visual Studio sotto Windows), tuttavia lavoro in maniera semplice e veloce.
Ripeto però, è una mia scelta personale. Ciascuno di Voi potrà operare scelte diverse dalla mia.
Ci sono ottimi IDE per lo sviluppo su C64. Abbiamo affrontato questo argomento sul numero "3" di "RetroMagazine". Invito quindi chi fosse interessato a conoscere tali strumenti a rileggere questo precedente articolo.

Circa il compilatore invece ho scelto KickAssembler tra diversi altri perché ne ho apprezzato la semplicità della sintassi, il trattamento dei files .sid, ma anche il fatto che è multipiattaforma.
Tuttavia non è open-source, mentre ad esempio lo è ACME. Quest'ultimo presenta una sintassi molto simile al KickAssembler oltretutto.
Esistono diversi altri compilatori, quali ad esempio: 64TASS, CC65, DASM, MXASS. Nel valutare un compilatore bisogna considerare diversi fattori.
E' sviluppato, aggiornato o no?
E' open-source?

Ha una sintassi semplice?
Offre strumenti quali macro, pseudo-opcodes, strumenti di debugging?
E' una analisi che può rivelarsi complessa. La volontà e l'esperienza che maturerete nel campo vi faranno rendere conto presto della bontà di un certo assemblatore rispetto agli altri.

Il codice che Vi presenterò quindi è compilabile con KickAssembler e dovrebbe essere facilmente modicabile per essere poi elaborato da qualsiasi altro compilatore di Vostro gradimento.

Parte IV – Il codice sorgente della intro commentato

Iniziamo subito ad esaminare il codice sorgente della intro:

```
.const mem_schermo=1024+40*10
.const chrout=$ffd2

.const debug=0 // impostare a 0 per rilasci finali. Diverso da 0 SOLO per eventuale debugging.

.label COUNTER0 = $ff
.label COUNTER1 = $ff

* = $0801 // BASIC start address (#2049)
BasicUpstart(codice)
        //
        // .byte $0b,$08,$0a,$00,$9e,$34,$30,$39,$36,$00
        //
        // Codice corrispondente alla linea BASIC: 10 SYS4096
        // per i compilatori che non creano automaticamente la linea per il lancio
        // del programma da BASIC.
        //
```

In questa prima parte, dichiaro mem_schermo, che mi servirà per fissare la zona di memoria di schermo video dove disegnare la scritta “RetroMagazine” tra due muri, uno superiore ed uno inferiore, quindi dichiaro la costante chrout al valore esadecimale $ffd2.
Chrout è una routine prevista dal Kernal del C64 che permette di scrivere sullo schermo. Vedrete il suo uso via via che proseguirò nel commentare questo codice.
Dichiaro poi una costante che ho chiamato debug, e la fisso a 0. Se la impostate ad un qualsiasi valore diverso da 0, vedrete a video la intro senza caratteri ridefiniti.
Fate attenzione ai commenti interni al codice (quando presenti), che si trovano subito dopo i simboli //.

Definisco poi due variabili, COUNTER0 e COUNTER1, che fanno riferimento alla stessa locazione di memoria, $ff ovvero la locazione di memoria 255, che ho scelto arbitrariamente.
Con *=$0801 definisco il punto della memoria dove inserirò il codice che si trova subito dopo.
BasicUpstart(codice) mi crea in automatico la linea BASIC 10 SYS4096 necessaria a lanciare la intro.
Subito dopo, nei commenti, i codici corrispondenti nel caso in cui il Vostro compilatore non permetta di fare lo stesso in automatico, codici che vanno comunque inseriti, byte per byte, a partire dalla locazione $0801, 2049 in notazione decimale.

```
.pc=$1000 "codice"
codice:
{
        jsr scr_init // imposta schermo, riproducendo il testo "RetroMagazine" tra due barre orizzontali piene

main:
        ldx COUNTER0

        lda sinusTable0,x //Recupera il nuovo valore di linea raster
        cmp $D012 //dalla relativa tabella, con indice COUNTER0, e lo confronta con l'attuale...
        bne no_barra0
        sei
        //disabilita le interruzioni
        jsr fai_barra0 //se corrispondente, crea barra raster tipo "0"
        cli
        //riabilita le interruzioni

no_barra0:
        ldx COUNTER1
        lda sinusTable1,x //Recupera il nuovo valore di linea raster
        cmp $D012 //dalla relativa tabella, con indice COUNTER1, e lo confronta con l'attuale...
        bne no_barra1
        sei
        //disabilita le interruzioni
        jsr fai_barra1 //se corrispondente, crea barra raster tipo "1"

        cli
        //riabilita le interruzioni

no_barra1:

        jmp main
}
```

Questo è il ciclo principale della intro.

Il codice parte dalla locazione $1000 (4096 in decimale), come si comprende dalla lettura di .pc=$1000 “codice”
Questo è un metodo ulteriore per fare quanto visto precedentemente con *=$0801.
Alcuni compilatori prevedono solo uno di questi due metodi e non entrambi.
Quindi la tag codice, che viene utilizzata da BasicUpstart(codice) discusso poc'anzi per determinare a quale locazione di memoria occorra associare la SYS di partenza.
Tra i simboli { e } troveremo tutte le istruzioni relative al ciclo principale della intro.
La jsr scr_init richiama una subroutine che si trova a partire dalla locazione scr_init (JSR=Jump to SubRoutine). Al termine di questa subroutine l'esecuzione del codice riprenderà dalla istruzione successiva.
Troviamo una label main e subito dopo l'istruzione ldx COUNTER0. Questa istruzione serve per riempire il registro X con il contenuto della locazione COUNTER0.
Quindi abbiamo l'istruzione lda sinusTable0,x.
Cosa significa questa istruzione?
Che a partire dalla locazione sinusTable0 ci spostiamo di un numero di bytes corrispondenti al valore contenuto nel registro X appena caricato, preleviamo il valore che troviamo all'interno di questa ben precisa locazione di memoria e lo memorizziamo infine all'interno dell'accumulatore.
Questo è uno degli usi più classici del registro X, che Vi dicevo essere un registro di indirizzamento.
Con l'istruzione cmp $d012 confrontiamo il contenuto dell'accumulatore con il valore contenuto nella locazione $d012, che Vi ricordo è la locazione che contiene i primi 8 bit del numero di riga che viene disegnata correntemente dal raster.
Se i due valori corrispondono, il flag Z del Registro di Stato del processore passa al valore logico '1' altrimenti tale flag vale 0.
L'istruzione seguente, bne no_barra0 (BNE=Branch on Not Equal, salta se non uguale) fa si che l'esecuzione del codice riprenda dalla label no_barra0 se il flag Z vale 0, ovvero se il contenuto dell'accumulatore non corrisponde al valore contenuto in $d012.
Se invece il flag Z vale 1, tale istruzione NON viene eseguita, e verranno eseguite invece le successive sei (SEI=SEt Interrupt), jsr fai_barra0, cli (CLI=CLear Interrupt) che servono rispettivamente per disabilitare le interruzioni, (argomento di cui non abbiamo ancora discusso ma che tratteremo nella prossima 'puntata'), lanciare l'esecuzione della subroutine fai_barra0 e, al termine dell'esecuzione di tale subroutine, vengono riabilitate le interruzioni.
A partire dalla label no_barra0 troviamo l'istruzione ldx COUNTER1. Quindi abbiamo l'istruzione lda sinusTable1,x che caricherà

all'interno dell'accumulatore il valore contenuto nella locazione di memoria a partire dalla label sinusTable1 e spostandoci poi in avanti del numero di locazioni di memoria corrispondenti al contenuto del registro X.
Con l'istruzione cmp $d012 confrontiamo il contenuto dell'accumulatore con il valore contenuto nella locazione $d012 (la locazione che contiene i primi 8 bit del numero di riga che viene disegnata correntemente dal raster) Se i due valori corrispondono, il flag Z del Registro di Stato del processore passa al valore logico '1' altrimenti tale flag vale 0.
L'istruzione seguente, bne no_barra1 fa si che l'esecuzione del codice riprenda dalla label no_barra1 se il flag Z vale 0, ovvero se il contenuto dell'accumulatore non corrisponde al valore contenuto in $d012.
Se invece il flag Z vale 1, tale istruzione NON viene eseguita, e verranno eseguite invece le successive sei, jsr fai_barra1, cli che servono rispettivamente per disabilitare le interruzioni, lanciare l'esecuzione della subroutine fai_barra1 e, al termine dell'esecuzione di tale subroutine, vengono riabilitate le interruzioni.
Infine, subito dopo la label no_barra1 troviamo l'istruzione jmp main (JMP=JuMP), che fa riprendere il ciclo a partire dalla label main.

Qual'è la logica di questo procedimento? Vi sono due aree di memoria,sinusTable0 e sinusTable1, che contengono due serie di numeri, risultato di due funzioni trigonometriche, seno e coseno.
Viene comparato il valore contenuto nelle aree di memoria sinusTable0(x) e sinusTable1(x) con il valore della riga disegnata correntemente dal raster.
Quando i valori coincidono, ecco apparire una rasterbar, di colore rosso oppure blu in corrispondenza di tale riga coincidente. Il diverso colore è dato dal fatto che il numero della riga coincida con ciò che è stato letto nell'area di sinusTable0(x), che farà ottenere una rasterbar rossa, oppure una rasterbar di colore blu nel caso in cui invece il numero di riga coincida con il valore letto nell'area sinusTable1(x).
Il contenuto della locazione $ff viene quindi incrementato, in maniera tale che il controllo successivo possa generare una rasterbar via via sempre più in basso sullo schermo, a partire dall'ultima posizione rilevata, finché ad un certo punto, superando il valore massimo (255, il numero più grande che è possibile memorizzare con bytes formati da 8 bit), il contenuto della locazione $ff torna a 0.
Il caratteristico andamento della funzione seno e della funzione coseno (analoga alla prima ma "sfasata" di un angolo di 90 gradi rispetto alla stessa), fa si che ci siano punti più ravvicinati nella zona superiore dello schermo ed in quella inferiore, mentre nella zona centrale dello schermo i punti si troveranno meno ravvicinati. Di conseguenza otterremo delle rasterbar più 'lente' nella zona superiore ed inferiore dello schermo e più veloci nella parte centrale.
Inoltre le due funzioni, grazie a questo sfasamento costante tra loro, ci permetteranno di ottenere delle rasterbars che si muoveranno indipendentemente l'una dall'altra e senza discontinuità. Noterere quando commenteremo il relativo codice che l'incremento del contenuto della variabile $ff, che sarà il contenuto del registro X utilizzato come indice per leggere i valori dei vari elementi di sinusTable0( ) e sinusTable1( ) avverrà solo all'interno della subroutine che visualizza la rasterbar. Ciò per assicurarci che ogni successiva uguaglianza riscontrata crei una rasterbar del medesimo colore in un'area prossima a quella dove è stata visualizzata appena prima.
E' superfluo aggiungere che le funzioni trigonometriche, in virtù di queste ed altre caratteristiche, sono spesso impiegate nelle intro.

```
//===================================
fai_barra0:
{
        ldy #10   //Tempo di attesa
idle1:  dey       //per minimizzare il "tremolio"
        bne idle1 //all'inizio dell'effetto

//-----------------------------------------------------
// Ciclo per stampare la barra raster '0'
//-----------------------------------------------------
        ldx #00

loop:   lda colorBar0
        //Assegna colore al bordo schermo
        sta $d020       //ed all'interno schermo
        sta $d021

        ldy #08   // Tempo di attesa
idle2:  dey       // per minimizzare
        bne idle2 // il tremolio alla fine dell'effetto
        //e rendere la barra più spessa...

        inx             //
        cpx #09         //
        bne loop        //

//-----------------------------------------------------
// Fine del ciclo
//-----------------------------------------------------

        lda #00         // Assegna il colore #00 (NERO)
        sta $d020       // al bordo
        sta $d021       // ed allo schermo video

        inc COUNTER0

        rts
}
//===================================
```

Questa è la subroutine che si occupa di fare apparire una rasterbar di tipo "0" ovvero una rasterbar di colore rosso.

Subito dopo la label fai_barra0 troviamo, racchiuso tra i due simboli { e } tutto il codice della subroutine. Tutto ciò che si trova tra le due parentesi graffe è all'interno di quello che si chiama tecnicamente un'area a visibilità locale e non può essere "visto", non può essere raggiungibile dall'esterno della stessa area.Notate che la label fai_barra0 si trova definita al di fuori. Perchè? Perchè viene richiamata da altre parti del codice, precisamente dall'interno della routine principale già discussa.Il vantaggio immediato di questa feature è che non è necessario preoccuparsi che le labels che puntano a diverse aree del nostro codice siano SEMPRE diverse, si chiamino SEMPRE in modo diverso. L'importante è che lo siano all'interno dell'area a visibilità locale dove le intendo usare.KickAssembler, insieme ad altri compilatori, permette di definire e gestire queste aree di codice 'stagne', tra loro.
In KickAssembler, ma anche con altri compilatori, è possibile all'occorrenza impiegare il sistema di numerazione binario oppure quello esadecimale oppure ancora quello decimale.
L'unica regola da rispettare è specificare un carattere davanti al valore numerico per esplicitarne la base numerica. Per un dato numerico in esadecimale, occorre il simbolo $ (esempio: #$80).
Per un dato binario, occorre invece il simbolo % (esempio: #%0111111).
Per un dato in base decimale non si antepone nulla davanti (esempio: #12).
Il simbolo # che vedete è necessario in quanto stiamo specificando valori numerici ben precisi, e non ci stiamo riferendo al contenuto di locazioni di memoria da leggere.

Esaminiamo adesso le prime tre righe della subroutine.
La prima istruzione, ldy #10, carica il valore 10 (in notazione decimale) nel registro Y.
La seconda istruzione, contrassegnata con la label idle1, dey, decrementa di 1 il contenuto del registro Y.
La terza, bne idle1 effettua un salto indietro, verso l'istruzione contrassegnata con la label idle1, se il decremento precedente non ha azzerato il contenuto del registro Y.

Se ci riflettete, partendo da un valore 10, saranno necessari 10 decrementi di 1 per azzerare il registro Y. Quindi, l'istruzione bne idle1 tornerà indietro verso l'istruzione contrassegnata dalla label idle1 10 volte, per effettuare poi un decremento di 1 del contenuto del registro Y per altrettante 10 volte affinché l'esecuzione possa proseguire successivamente.
A cosa serve quindi tutto ciò? Semplice, a PERDERE TEMPO. Giusto un po', per limitare quel leggero "tremolio" della rasterbar che a volte purtroppo noterete comunque.
E' possibile calcolare con esattezza quanto tempo stiamo perdendo, ma lascio tali calcoli ai puristi e/o esperti in materia :)

Passiamo ora a cosa succede subito dopo.
Con l'istruzione ldx #00 carichiamo dentro il registro X il valore numerico 0.
L'istruzione contrassegnata con la label loop, lda colorBar0 carica un ben preciso valore nell'accumulatore. Se scorrete il codice, troverete che il valore all'indirizzo colorBar0 è 02, il codice che esprime il colore ROSSO.
Depositiamo il contenuto dell'accumulatore quindi all'interno di due locazioni di memoria, $d020 e $d021, rispettivamente 53280 e 53281 in notazione decimale, che, come molti di Voi ricorderanno, sono le locazioni che controllano il colore del bordo e dello schermo del C64.
Quindi in questo preciso momento stiamo colorando bordo e schermo del C64 in ROSSO!
Le tre istruzioni successive servono a PERDERE ANCORA ALTRO TEMPO, secondo la stessa logica già discussa.
Precisamente, ldy #08 carica nel registro Y il valore decimale 8.
L'istruzione contrassegnata con la label idle2 decrementa il contenuto attuale del registro Y.
L'istruzione successiva, bne idle2, torna indietro e continua a decrementare di 1 il contenuto del registro Y finché esso non si azzererà.
Perchè introdurre questo ritardo? Perchè vogliamo ottenere una barra rossa ben visibile, sufficientemente larga.Tuttavia questo ritardo non basterà. Ne introduciamo uno ulteriore.
L'istruzione successiva, inx, incrementerà di 1 unità il contenuto del registro X.
L'istruzione cpx #09 confronterà il contenuto attuale del registro X con il valore numerico 9 in notazione decimale. Se i due valori coincideranno il flag Z del registro di stato del processore, come già discusso, passerà al valore logico '1', altrimenti varrà '0'.
L'istruzione bne loop farà proseguire l'esecuzione del codice dalla locazione contrassegnata con la label loop finché i successivi decrementi del contenuto del registro X non lo abbiano azzerato.
Abbiamo realizzato quello che in gergo tecnico si chiama iterazioni nidificate. Stiamo osservando un ciclo di ritardo all'interno di un altro ciclo di ritardo, più esterno.
Lo scopo è quello, come detto sopra, di ottenere una barra rossa ben visibile, ben spessa. Ecco perché occorre perdere un tempo sufficientemente lungo per mantenere il colore dello schermo e del bordo di colore rosso.
Finalmente, dopo un ritardo ampio a sufficienza, possiamo far tornare lo schermo ed il bordo al colore di partenza scelto, che è il colore nero. Come? Con le semplici istruzioni seguenti:
lda #00, sta $d020, sta $d021.
La prima carica il valore 0 all'interno dell'accumulatore, la seconda copia il contenuto dell'accumulatore all'interno della locazione $d020 (53280 in decimale), la terza copia il contenuto dell'accumulatore all'interno della locazione $d021 (53281 in decimale), locazioni che dovremmo ormai ben conoscere.

Infine, prima di uscire dalla subroutine per ritornare alla routine principale con l'istruzione rts (ReTurn from Subroutine), tramite l'istruzione inc COUNTER0 incrementiamo di 1 il contenuto della locazione COUNTER0.

La subroutine che si occupa di creare la rasterbar blu è identica, nella logica e nelle istruzioni, a questa discussa.
Le uniche differenze sono che, invece di caricare il valore contenuto nella locazione colorBar0 (Rosso come dicevamo prima), andremo a caricare nell'accumulatore il contenuto della locazione colorBar1, che contiene invece il valore 06, che corrisponde al colore Blu ovviamente.
La seconda ed ultima differenza è che prima della rts per tornare alla routine principale, incrementeremo di 1 la locazione COUNTER1 invece della locazione COUNTER0. Quest'ultima è una differenza solo formale, non sostanziale in quanto le label COUNTER0 e COUNTER1 puntano entrambe alla medesima locazione di memoria, $FF ovvero 255 in decimale.

Non ci resta che commentare il funzionamento della subroutine scr_init che si occupa di ripulire lo schermo e di scrivere i caratteri ridefiniti corrispondenti ai due muri ed alla scritta "RetroMagazine".

```
scr_init:
{
        lda #$00
        sta $d020  // Bordo NERO
        sta $d021  // Schermo NERO

        lda #147
                // Pulisce lo schermo video,
        jsr chrout
                // chiamando la
                // famigerata routine
                // kernal chrout con
                // argomento 147.


if (debug==0)
{
        //////////////////////////////////////////////////
        // Abilita la modalità schermo con set di caratteri ridefiniti

        .const screen=$0400
                // locazione memoria di schermo video
        .const charset=$3000
                // locazione memoria set di caratteri

        lda #[[screen & $3FFF] / 64] | [[charset & $3FFF] / 1024]     //

        sta $D018          //
}
```

Nelle prime 3 righe di scr_init, impostiamo a 0 l'accumulatore e lo memorizziamo dentro le due locazioni $d020 e $d021, ovvero colore bordo e colore schemo. Risultato? Schermo e bordo entrambi di colore nero, ovviamente.

Carichiamo poi nell'accumulatore il valore #147 e richiamiamo la routine chrout del Kernal, routine ben conosciuta che stampa sullo schermo l'attuale contenuto dell'accumulatore.
Il risultato sarà quello di ottenere uno schermo pulito, senza alcun carattere visualizzato, risultato analogo a quello che otterremmo digitando in BASIC print chr$(147); e dando poi invio.

Successivamente, all'interno del gruppo di istruzioni relativo alla istruzione .if (debug==0) impostiamo il corretto valore dentro la locazione $d018 (53272 in decimale) per ottenere la memoria video a partire dalla locazione $0400 (1024 in decimale) e set di caratteri ridefiniti a partire dalla locazione $3000 (12288 in decimale).
Il calcolo del valore corretto viene effettuato comodamente tramite la formula #[[screen & $3FFF] / 64] | [[charset & $3FFF] / 1024].
Non mi dilungherei troppo su questo punto...l'importante è che faccia bene il lavoro che deve fare :)

Una nota importante,invece qui è necessaria. All'inizio dei commenti al codice Vi dicevo di lasciare sempre il valore debug impostato a 0 proprio perché in caso contrario tutto il blocco di istruzioni relativo a questa istruzione .if(debug==0) verrebbe saltato a piè pari se debug non fosse in effetti uguale a 0. La conseguenza immediata sarebbe che NON verrebbe impostato il corretto valore nella locazione 53272 e non vedremmo quindi i caratteri con la "nuova" veste grafica che ho realizzato appositamente per questa intro.

Veniamo adesso alla parte successiva del codice restante.

```
        ldx #00
s0:
        lda #$71 // Riempie la memoria video con il valore $71 (#143) e $72 (#144)
        sta $0400,x     // che corrisponde al carattere 'muro'...
        sta $0500,x     //
        lda #$72
        sta $0600,x     //
        sta $06e8,x     //

        lda #$01 // Setta il colore dello schermo al valore '01'
        sta $d800,x     // ovvero BIANCO
        sta $d900,x     //
        sta $da00,x     //
        sta $dae8,x     //

        inx             // Incrementa x
        bne s1
```

Come potrete vedere aiutandoVi anche leggendo i commenti inseriti insieme al codice, azzero il contenuto del registro X, quindi carico il valore #$71 e lo memorizzo a partire dalle locazioni $0400 e $0500 con indice il contenuto del registro X.
Faccio lo stesso con il valore #$72, riempiendo stavolta le locazioni a partire da $0600 e $06e8, sempre con indice il contenuto del registro X.
Questi 2 codici carattere corrispondono entrambi al carattere 'muro' che vedrete sopra e sotto la scritta "RetroMagazine".
Carico infine il valore #$01 (che corrisponde al colore dello schermo BIANCO) e lo memorizzo nelle locazioni a partire da $d800, $d900, $da00 e $dae8 con indice il contenuto del registro X.
Incremento di 1 unità il registro X e finché il suo contenuto non si azzererà nuovamente (dopo quindi 256 incrementi) riprendo l'esecuzione del codice a partire da s1 tramite l'istruzione bne s1 che ormai conoscete bene!

Il risultato sarà che lo schermo sarà riempito con un muro apparentemente unico, creato però con due caratteri diversi (di codice #$70 e #$71 precisamente), ed il colore di questo "muro" sarà chiaramente bianco!

```
        ldx #39
        lda #$03
s1:
        sta $0400+40*0,x
        sta $0400+40*1,x
        sta $0400+40*2,x
        sta $0400+40*3,x
        sta $0400+40*4,x
        sta $0400+40*5,x

        sta $0400+40*20,x
        sta $0400+40*21,x
        sta $0400+40*22,x
        sta $0400+40*23,x
        sta $0400+40*24,x
        sta $0400+40*25,x
        dex
        bpl s1
```

Qui rimuoviamo le parti di "muro" in eccesso sullo schermo.
Stavolta, carichiamo il registro X con il valore decimale 39, quindi carichiamo nell'accumulatore il valore #$03 che corrisponderà ad un carattere appositamente "vuoto" e riempiremo così la riga 0, 1,2,3,4,5 ed ancora, le righe 20,21,22,23,24,25 dello schermo video.
Decrementiamo il contenuto del registro X di una unità e finché il suo valore sarà positivo riprenderemo l'esecuzione del codice a partire da s1 tramite l'istruzione bpl s1 (BPL=Branch on Plus)

```
        // DISEGNA BARRA SUPERIORE
        ldx #39
        lda #$70
s2:     sta mem_schermo-40,x
        sta mem_schermo-40*5,x
        dex
        bpl s2
```

In questa porzione di codice delimitiamo la parte di muro superiore. Carichiamo il valore decimale 39 nel registro X, quindi il valore esadecimale $70 nell'accumulatore e riempiamo le locazioni a partire da mem_schermo-40 e mem_schermo-40*5 con il valore contenuto nell'accumulatore, spostandoci tramite il contenuto del registro X all'interno delle due righe di schermo,da sinistra verso destra. Decrementiamo il contenuto del registro X e finché il suo valore sarà positivo riprenderemo l'esecuzione del codice a partire da s2 tramite l'istruzione bpl s2, come visto appena sopra.

Vi ricordo che mem_schermo è una costante che è stata definita all'inizio del codice e vale precisamente 1024+40*10.

```
// DISEGNA SCRITTA RETROMAGAZINE
        ldx #0
ancora:
        lda map_data,x
        sta mem_schermo,x
        inx
        cpx #40*6
        bne ancora
```

In questa porzione di codice scriviamo "RETROMAGAZINE" nella zona centrale dello schermo.

Stavolta, invece di usare un valore fisso, costante, per riempire lo schermo utilizzeremo un altro sistema.
Caricheremo i diversi valori che corrisponderanno ai vari caratteri che comporranno insieme la scritta "RETROMAGAZINE" a partire dalla locazione map_data nell'accumulatore, utilizzando come indice il contenuto del registro X, azzerato inizialmente.
Verseremo poi ciascuno di questi caratteri a partire dalla locazione mem_schermo, con indice sempre il contenuto del registro X.

Incrementiamo, dopo aver letto e scritto ciascun carattere, il contenuto del registro X di una unità.

Dal momento che intendiamo riempire ben 6 righe di schermo in questo modo, confrontiamo il contenuto del registro X con il valore #40*6 (ovvero 240) e finché sarà diverso, riprenderemo l'esecuzione del codice a partire dalla tag ancora finché avremo caratteri da piazzare ancora sullo schermo.

Il codice per delimitare il muro inferiore, analogo a quello già visto prima, lo copio semplicemente:

```
        // DISEGNA BARRA INFERIORE
        ldx #39
        lda #$70
s3:     sta mem_schermo+40*6,x
        sta mem_schermo+40*10,x
        dex
        bpl s3
```

Notate che cambiano soltanto le righe di schermo che andremo a riempire con il carattere $70 (che corrisponde chiaramente ad un carattere completamente pieno), come era facile intuire, naturalmente.

A questo punto, nella parte restante del codice sorgente, ci saranno le parti necessarie al funzionamento corretto dello stesso, ovvero:

```
*=* "dati"

colorBaro:
.byte 02

colorBar1:
.byte 06

sinusTableo:
.fill 256, 150 + round(110 * sin(toRadians(360
* i / 256)))

sinusTable1:
.fill 256, 150 + round(110 * cos(toRadians(360
* i / 256)))

.pc=$3000 "charset_data"
.import binary "retromagazine_demo.bin"

map_data:
.byte
$00,$01,$02,$03,$03,$03,$03,$03,$03,$03,$0
3,$04,$03,$05,$03,$03,$03,$03,$03,$03,$03,$
03,$03,$03,$03,$03,$03,$03,$03,$03,$03,$03,
$03,$03,$03,$03,$03,$03,$03,$03
.byte
$06,$07,$08,$09,$0a,$0b,$0c,$0d,$0e,$0f,$1
0,$11,$12,$13,$14,$15,$00,$16,$17,$18,$19,$
1a,$1b,$1c,$1d,$1e,$1f,$20,$21,$03,$03,$03,
$03,$03,$03,$03,$03,$03,$03,$03
.byte
$06,$22,$23,$24,$25,$26,$27,$28,$29,$2a,$2
b,$2c,$2d,$2e,$2f,$30,$31,$32,$33,$34,$35,$
36,$37,$38,$39,$3a,$3b,$3c,$3d,$3e,$03,$03,
$03,$03,$03,$03,$03,$03,$03,$03
.byte
$06,$3f,$40,$41,$42,$43,$44,$45,$46,$47,$4
8,$49,$4a,$4b,$4c,$4d,$4e,$4f,$50,$51,$52,
$53,$54,$55,$39,$56,$57,$58,$59,$03,$03,$0
3,$03,$03,$03,$03,$03,$03,$03,$03
.byte
$5a,$5b,$5c,$5d,$5e,$03,$5f,$60,$03,$61,$6
2,$63,$64,$65,$66,$67,$68,$69,$6a,$6b,$6c,
$5f,$6d,$5d,$5d,$6e,$6f,$61,$67,$03,$03,$03
,$03,$03,$03,$03,$03,$03,$03,$03
.byte
$03,$03,$03,$03,$03,$03,$03,$03,$03,$03,$0
3,$03,$03,$03,$03,$03,$03,$03,$03,$03,$03,$
03,$03,$03,$03,$03,$03,$03,$03,$03,$03,$03,
$03,$03,$03,$03,$03,$03,$03,$03
```

I colori delle due barre, in corrispondenza di colorBaro e colorBar1;

Lo sviluppo delle due funzioni sin(x) e cos(x), in corrispondenza di sinusTableo e sinusTable1;
Le due istruzioni per caricare in memoria il set di caratteri ridefinito, a partire dalla locazione $3000;
Infine, i dati che corrispondono a ciascun carattere che andrà a formare la scritta "RetroMagazine" a video.

Parte V – Compilare che passione…

Dopo aver commentato, cercando nella maniera più semplice possibile di spiegare il funzionamento del codice di questa intro, due parole su come sfruttare tale codice.

Il codice sorgente dovrà essere dato "in pasto" al compilatore, che produrrà la versione eseguibile di questa intro, lanciabile poi con il Vostro emulatore preferito oppure direttamente sul "commie" previo trasferimento del codice eseguibile nello stesso.

Il compilatore dovrà essere KickAssembler, a meno che vogliate adoperare un altro compilatore di Vostro gradimento, correggendo però opportunamente il codice sorgente perché così come lo vedete è utilizzabile senza effettuare alcuna correzione con KickAssembler.

Quindi i passaggi sono i seguenti:

1) Scaricate e scompattate il KickAssembler in una cartella apposita: (http://www.theweb.dk/KickAssembler/KickAssembler.zip)

2) Scaricate e scompattate il codice di questa intro: (https://github.com/marcus73/retromagazine_01/archive/master.zip)

3) Modificate i percorsi contenuti nei files go.bat (sotto Win) e Makefile (sotto Linux) in maniera tale che puntino in maniera corretta a ciò che avete installato sul vostro computer, PRIMA di lanciare la compilazione.

4) Aprite la riga comandi, sotto Windows o sotto Linux, spostatevi all'interno della cartella contenente il codice della intro e mandate in esecuzione il file batch go.bat (sotto Windows) oppure potrete dare il comando make sotto Linux.
Se avete preparato tutto per bene vedrete partire il Vostro emulatore che eseguirà la intro.

Per i più pigri, fornisco comunque il codice della intro già compilato. Basterà aprire il file demo.d64 con il Vostro emulatore preferito e lanciare il primo (ed unico) file presente all'interno di questa immagine di disco c64.

E' tutto amici, almeno per questa prima "puntata".

Nella prossima cercheremo di capire meglio perché le bande sono un po' sfrangiate, ed aggiungeremo altri elementi alla intro, ad esempio un bel brano .sid eseguito in sottofondo.

Alla prossima!

Amiga – Il Chipset

di Dante Profeta

Amiga – l'Hardware

A volte è strano il caso. A volta sembra un cerchio, che ci ripiega su eventi o fatti già accaduti, sopiti, quasi archiviati in un passato remoto da sembrarci estraneo. Eppure si era là, partecipi, protagonisti diretti, attori di un presente pieno di stimoli crescenti. È così che vivevo la scena Amiga, come attore in prima persona, e come tale, dovevo scrivere un videogioco per essere parte del momento. Era questo l'obiettivo ultimo da raggiungere. Scrivere un videogioco. Il mio tempo era studio e ricerca, tentativi di hacking dell'hardware e del software, volto a trovare conferma in quanto appreso e spesso sottinteso nei manuali e negli articoli tecnici. Eccomi dunque un tempo a scrivere e sognare Agricola. Eccomi oggi qua, anche abbastanza incredibilmente per me, a rimembrare e rivivere con voi ciò che fu fonte di emozione e giubilo, e che per moltissimi di noi ha rappresentato il sogno di avere una sala giochi in casa, ma con una spinta propulsiva deflagrante: Amiga.

Amiga era la piattaforma, la base di lancio e l'astronave. Agnus, Paula e Denise erano un dono per tutti noi, una creazione d'ingegno, di sicuro partorite in uno dei più bei momenti di massima ispirazione tecnico-poetica. Si, perché nulla ha mai più suscitato tanta emozione e tanta euforia quanto Amiga nel mondo consumer, almeno a livello emotivo, per quelli della mia generazione.

L'idea era semplice: progettare un sistema basato su un microprocessore a 16 bit. Non so perché R.J. Miner scelse il Motorola 68000 a 32 bit, ma fu la prima di una delle più felici scelte progettuali. Altre vennero appresso. Famosa fu quella della modalità HAM. Oggi che mastichiamo bene o male quasi tutti l'inglese e siamo portati a tradurne i termini, diremmo: modalità prosciutto… Beh, ma se è per questo anche ai tempi ci scherzavamo sopra.

Denise

4096 colori! Tutti assieme?

HAM, Hold-And-Modify. La modalità HAM, che consiste nel variare una singola componente cromatica, o R il rosso, o G il verde, o B il blu, tra pixel adiacenti, avrebbe permesso la rappresentazione grafica su schermo di un'immagine di ben 4096 colori, mostrando al mondo l'intera bellissima palette a 12 bit di Amiga; 4 bit per il verde, 4 bit per il rosso, 4 bit per il blu.

Eh sì, perché sebbene Amiga avesse una palette a 4096 colori, poteva visualizzarne contemporaneamente solo un massimo di 32. Certo, scelti e composti a caso tra i 4096 colori disponibili, ma sempre e soltanto 32 colori contemporaneamente poteva visualizzare.

HAM fu un lampo di genio avuto da Miner durante un viaggio per andare a vedere un simulatore di volo. Fu proprio durante il viaggio che si rese conto che in natura non c'è un brusco cambio di colore tra un punto percepito con l'occhio e quello adiacente, ma che le gradazioni cromatiche sono morbide. L'acquisizione di tale consapevolezza, trasposta in Denise ha anche ovviato a un gravoso problema dell'epoca: la quantità di memoria necessaria per contenere un'immagine a 12 bit. Tenendo dunque traccia della variazione di una sola componente cromatica, ovvero 4 bit, rispetto al pixel adiacente, si è ridotto lo spazio richiesto in memoria per visualizzare l'immagine. In realtà, non si riduce ad 1/3 come si sarebbe indotti a calcolare, ovvero 12 bit diviso 4 bit, ma della metà, perché i bit coinvolti per mantenere l'informazione del pixel sono 6. 6 per ogni pixel, così divisi: I primi due bit prendono il nome di bit di controllo, e i restanti 4 bit portano con sé l'informazione sul colore.

Se i due bit di controllo sono posti a 0, allora i restanti 4 bit agiscono come registro indice indicando quale dei 32 colori della palette prescelta utilizzare; le restanti 3 configurazioni dei due bit di controllo indicano quale componente cromatica verrà modificata, rossa, verde, o blu, e i quattro bit di informazione del colore porteranno con se proprio la tonalità cromatica del colore prescelto. Ricordate? La palette è a 12 bit, 4 bit per il verde, 4 bit per il rosso, e 4 bit per il blu.

La prima versione di Denise era mancante di una modalità grafica, la modalità Extra Half-Bright (EHB), e I primissimi Amiga (1000) che vennero commercializzati, per fortuna nella sola versione americana NTSC, ne erano pertanto carenti; Famosa è l'intro del videogioco Full Contact di Team16, che sfrutta appunto la modalità grafica Extra Half-Bright.

Full Contact – Team 17

I bitplane

La grafica di Amiga si basava sui bitplane, piani di bit. Un bitplane è grande quanto tutto lo schermo, e in realtà talvolta più dell'area visibile, ma ci torneremo in seguito su questo punto, a ogni bit corrisponde un pixel monocromatico, che può essere acceso, se posto a 1, o spento, se posto a 0.

Con un singolo bitplane si può dunque visualizzare un'immagine monocromatica. Sovrapponendo però due bitplane si può visualizzare invece un'immagine a 4 colori,

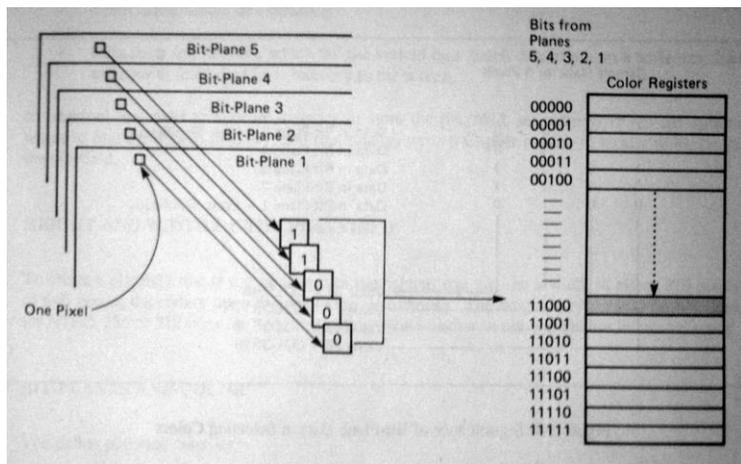
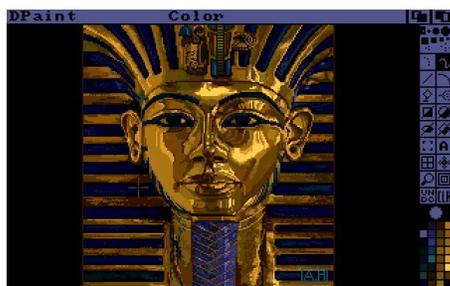



Immagine esplicativa presa dal "Amiga Hardware Reference Manual

giocando con le 4 configurazioni possibili dei due bit sovrapposti. In realtà l'immagine sarebbe a 3 colori, perché la configurazione 00 rappresenta la trasparenza e mostra il colore settato di sfondo.

Continuando così, con 3 bitplane l'immagine può avere 8 colori; con 4 bitplane possiamo avere un'immagine a 16 colori; e infine con 5 bitplane l'immagine può avere fino ai 32 colori, il massimo visualizzabile contemporaneamente. Ecco come vengono prodotte le immagini su Amiga.

Supponendo di avere un'immagine a 32 colori, in memoria verranno allocati 5 bitplane, in zone differenti, anche non contigue, e indirizzati tramite gli appositi registri BPLxPTH e BPLxPTL.

La maschera Tutankhamen – 32 colori con Deluxe Paint

La modalità Extra Half-Bright (EHB)

In realtà esiste un sesto bitplane, che però non concorre ad aumentare il numero di colori, ma a dimezzarne la luminosità.

Ricordate che ogni bitplane è una mappa di bit e che ogni bit può essere posto a 1 o a 0, e che identifica uno specifico pixel dell'immagine, vero? Ebbene, se il bit del sesto bitplane è posto a 1 allora la luminosità del colore del pixel corrispondente verrà dimezzata.

Da qui il nome Extra Half-Bright: Extra, per il sesto bitplane; Half-Bright, mezza tinta, per la luminosità dimezzata del pixel.

I playfield

I bitplane così composti venivano raggruppati sotto il nome di Playfield. Un playfield poteva dunque essere composto da 1 a 5 bitplanes, oppure 6 per la modalità EHB.

Invero Amiga permetteva di avere fino a due playfield. Due raggruppamenti di bitplane distinti, ma con un massimo di 3 bitplane per playfield.

Il dual playfield

Vi siete certamente chiesti come mai Amiga potesse realizzare quei fluidissimi scrolling parallattici, vero? Ebbene, semplice, con la tecnica del dual playfield.

Immaginate i due playfiled indipendenti come due immagini grafiche sovrapposte. Un'immagine su un playfield, e l'altra immagine sull'altro playfield; uno di fronte all'altro. I playfield giacciono dunque uno davanti e uno dietro, dove il colore con indice 0, tra quelli rappresentabili, del playfield anteriore non è un colore ma la trasparenza per mostrare il playfield posteriore, e il colore 0 del playfiled posteriore è anch'esso indice di trasparenza e mostra il colore settato di sfondo.

Agricola: Livello 1 - immagine nel Playfield Foreground

Agricola: Livello 1 - immagine nel Playfield Background

Agricola: Livello 1 - immagine completa con entrambi i Playfields e gli effetti raster col Copper

Avendo a disposizione 6 bitplane, in modalità dual playfield, il massimo raggruppamento che possiamo fare per singolo playfield è di 3 bitplane. Cosicché per playfield potremo avere al più 7 colori… ricordate? Il colore 0 è la trasparenza… ma niente panico, perché poi chiederemo aiuto ad Agnus per cambiare al volo i colori, visualizzandone quanti ne vorremo, sempre nel limite dei 4096 disponibili.

Quale dei due playfield mostrare di fronte e quale mostrare dietro è una mera scelta del programmatore, che può cambiare idea in qualsiasi momento della scansione orizzontale del pennello elettronico, e invertire le cose al volo, creando effetti grafici di assoluta bellezza se sfruttati opportunamente durante il gioco.

Si, perché si può decidere di avere ad esempio per le prime 50 righe di scansione il playfield 0 davanti e quello 1 dietro, e poi invertirli, e poi invertirli ancora, riga per riga.

Ma si possono anche mischiare tutte le modalità grafiche ad ogni riga di scansione orizzontale, così da avere ad esempio un gioco dual playfield dall'alto al basso, e sotto un bel pannello informativo con un singolo playfield a 32 colori o addirittura in HAM.

Fantastico vero? Tutto gestito dall'hardware! Potenza di Agnus e Denise, che concorrono allo scopo.

Science of Cambridge MK14, l'antenato dei Sinclair ZX

di Alberto Apostolo

Prima della Sinclair Research e del QL, dello ZX Spectrum e dello ZX81, prima ancora della Sinclair Computers e dello ZX80, esisteva la Scien-ce of Cambridge e il suo Microprocessor Kit 14 venduto in scatola di montaggio.

Si poteva ordinare solo per posta, come indicato negli avvisi pubblicitari (i primi dei quali risalgono al Febbraio 1978, quando fu messo in commercio).

L'idea non era nuova; il mercato offriva gia' agli appassionati di elettronica, la possibilita' di assemblare computer didattici fai-da-te a basso costo.

Nella tarda estate del 1977 Clive Sinclair si interesso' al progetto di un micro-elaboratore venduto in kit di montaggio, proposto da Ian Williamson (un ingegnere elettronico, impiegato alla Cambridge Consultants Ltd, una azienda che aveva gia' avuto in passato relazioni commerciali con Sinclair).

Chris Curry (uno dei manager della Science of Cambridge) fu incaricato di seguire il progetto e cominciarono i primi dissapori. Curry chiese a Williamson di apportare modifiche al progetto originale e di usare esclusivamente componenti elettronici prodotti dalla azienda americana National Semiconductor, la quale realizzava il microprocessore SC/MP (quello scelto da Williamson per il kit).

Mentre Clive Sinclair, presumendo di avere tutte le informazioni possibili per arrivare a un prodotto commerciabile, intendeva estromettere Ian Williamson dalla collaborazione, rivedendo gli accordi economici stabiliti in precedenza (Williamson stava lasciando la Cambridge Consultants Ltd per trasferirsi alla British Leyland).

Alla fine il progetto originale di Williamson fu modicato distaccandosi dall'utilizzo di una calcolatrice per usare interfacce piu' convenzionali per tastiera e display e nacque il MK14.

Curry inoltre sperava di poter vendere, in aggiunta, espansioni per poterlo trasformare in un computer completo. Allo scopo aveva immaginato una interfaccia per permttere al kit di connettersi a un televisore domestico, pensava di dotare il sistema di un interprete BASIC da caricare con un registratore a cassette (visto che tutti ne avevano uno a casa). Aveva per fino in mente di mettere a disposizione una specie di schermo-tastiera comandato da una penna ottica.

Figura 2 - Rappresentazione del MK14 (immagine presa da https://nosher.net) .

Il primo prototipo del MK14 fu costruito da Steve Furber (all'epoca un laureato in matematica, appassionato della tecnologia dei microprocessori, che seguiva a Cambridge un dottorato in ingegneria aeronautica).

Egli realizzo' il circuito saldando le parti nel suo salotto di casa, grazie allo schema elettrico che gli era stato consegnato. Secondo alcuni, in pratica il MK14 era una copia del sistema di sviluppo SC/MP Intro-Kit venduto dalla National Semiconductor. Secondo altri, non era una semplice riproduzione del SC/MP. Infatti, sempre attento ai costi, Chris Curry aveva fatto in modo che la scheda facesse uso di poche (economiche) parti, tra cui la tastiera.

Figura 1 -Sistema di sviluppo Intro-Kit.

Il prototipo aveva due PROM che contenevano la copia della ROM gia' scritta di un SC/MP. Ma la copia non era stata fatta bene e Furber dovette sobbarcarsi anche il debug delle PROM. La realizzazione del prototipo fu il suo primo incarico alla Science of Cambridge.

Dopo che gli errori furono corretti, lo schema elettrico fu riprodotto come circuito stampato e il MK14 finalmente messo in vendita per corrispondenza nel Febbraio 1978.

Il prezzo di vendita fissato era di 39.95 Sterline piu 3.20 Sterline di VAT4 e 40 pence di P&P.

Una volta assemblato (Figura 2), l'utente poteva caricare i propri programmi attraverso la tastiera esa-decimale (seguendo passo per passo le indicazioni mostrate nel manuale fornito con il MK14).

Furono necessari almeno tre mesi affinche' le vendite raggiungessero livelli apprezzabili. Ma poi gli ordini da parte di compratori interessati (20mila) superarono di gran lunga la previsione iniziale di 2mila.

Il successo poteva essere attribuito sia al prezzo di vendita molto basso, sia all'aspetto professionale della pubblicita' di Curry, assai distante da quelle degli altri micro-elaboratori in kit e i nuovi micro-computer che dovevano fare ancora la loro comparsa (come il Nascom 1).

Al potenziale acquirente, le pubblicita' accorte di Curry facevano apparire il MK14 come un prodotto serio. Tuttavia, come facevano notare i recensori delle riviste di

informatica ed elettronica, non mancavano i difetti.

Clifford Clark (nel numero di Marzo 1979 di Personal Computer World) riportava:- Il primo difetto, la tastiera, e' piu' o meno considerato dai costruttori in quanto essi forniscono un connettore piatto (edge connector) sul circuito stampato per collegare una tastiera esterna...

...Il secondo difetto e' il manuale, il quale salta dal semplice switching della memoria a programmi di esempio specificati in formato assembler. Ci sono istruzioni riguardo la scrittura sul display senza nessuna spiegazione su come si faccia.-

La tastiera del MK14 era una delle componenti piu' delicate. Essa sembrava derivare da un'altro sistema di sviluppo in kit della National Semiconductor, progettato prima dell'Intro Kit e chiamato Low-Cost Development System (L.V.D.S.). Era una grossa scatola nera nella quale un certo numero di schede poteva essere montato, una per il processore SC/MP e altre per la memoria, le unita' di I/O ecc. Aveva una tastiera esa-decimale esattamente come fu usata piu' tardi sul MK14. Ma la tastiera non era resistente e spesso si guastava.

Ian Williamson avrebbe preferito vendere il MK14 a 49,99 Sterline (un buon prezzo lo stesso, secondo lui) e fornire una tastiera di migliore qualita', ma fu deciso diversamente.

Non si sa chi scrisse il manuale allegato con il MK 14 stesso, quello che Clifford Clark e i recensori di altre testate criticavano. Molto probabilmente fu scritto da qualcuno del crescente numero di appassionati di elettronica e informatica dell'Universita di Cambridge che avevano a che fare con Chris Curry. Certamente si sa che uno di loro, Davis Johnson-Davies, realizzo' i programmi di esempio inseriti nel manuale.

Comunque se la tastiera aveva un costo ridotto al punto che si rompeva dopo poco tempo, il resto del sistema resisteva bene ed era funzionale agli scopi per i quali era stato progettato (imparare a programmare microprocessori con poca spesa).

Infine vi era la scarsa capacita di Sinclair di gestire gli ordini. Come accadde molto piu tardi con i computer Sinclair, il MK14 soffriva lunghi tempi di consegna.

Chris Curry aveva ordinato un numero di chip insufficiente a fronte della inaspettata quantita di ordini ricevuti e ci vollero parecchi mesi per l'arrivo di altri componenti. Di conseguenza non ebbe mai le quantita' giuste e i ritardi di consegna afflissero il prodotto per il resto del 1978 e per il 1979.

Nel frattempo, il MK14 divenne un sistema modulare e numerose schede aggiuntive permisero di espandere il sistema: una interfaccia per registratore a cassette, integrati di espansione RAM a 128 e a 256 byte da aggiungere per avere un totale di 640 byte.

Nelle vendite si avvicendarono 5 versioni che differivano l'una dall'altra per la RAM aggiuntiva, gli integrati di I/O, la tastiera (passando da quella a membrana a quella meccanica, al fine di migliorarne l'affidabilita). Qualche tempo dopo, Clive Sinclair decise che i computer erano il modo giusto per fare soldi e inizio un nuovo progetto : un computer completo che costasse meno di 100 Sterline, ossia lo ZX80 (Figura 3).

Figura 3 - Il Sinclair ZX 80.

Dopo lo ZX80 arrivarono i modelli successivi della serie ZX (ZX81, ZX Spectrum, ecc.) nei quali si potevano notare alcuni dei "peccati originali" del MK14 come la famigerata tastiera a membrana, la componentistica a basso costo, i lunghi tempi di consegna dei prodotti. Tuttavia queste limitazioni non hanno impedito a questo prodotto e ai computer della serie ZX di riscuotere un grande successo di pubblico. Un suc cesso che ha garantito a Clive Sinclair di ricevere dalla Regina Elisabetta II il titolo di Sir nel 1982.

Procurarsi un MK14 e' estremamente difficile. Essendo stato prodotto in circa 50mila esemplari (nemmeno tutti assemblati completamente), e' considerato una rarita', una sorta di Santo Graal dell'informatica britannica per gli appassionati e i collezionisti di cimeli informatici. Su Ebay, in passato sono avvenute vendite all'asta di esemplari di MK14 con valutazioni tra 500$ e 800$. Altri appassionati, hanno invece provato a realizzare dei cloni e cercando in Rete si possono trovate progetti di un certo interesse come per esempio quello di Paul Robson (http://mymk14.co.uk/paulRobson/index.htm%7D) oppure il progetto di Colin Phillips ([Gra10]) con il suo MK14 3.0 dove l'originale microprocessore del MK14 e stato sostituito con un controller Microchip PIC.

Bibliografia

[Ber79] A.A. Berk, Science of Cambridge MK 14, Practical Electronics, Maggio 1979, http://s400081762.websitehome.co.uk/paulRobson/review.htm .

[FP83] V. Falzone, G. Pompei, Elaboratori elettronici e loro applicazione, Calderini, 1983.

[Gra10] J. Grahame, Rebirth Of The First Sinclair Computer, http://www.retrothing.com/2010/06/rebirth-of-the-first-sinclair-computer.html.

[MT87] F. Mariotti, G. Torino, Sistemi, automazione e laboratorio, vol. 2, Piccin, 1987.

[OCM17] Old Computer Museum, http://www.oldcomputermuseum.com/mk-14.html.

[Smi14] T. Simith, Ian Williamson: The engineer who gave Sinclair his rst micro, https://www.theregister.co.uk/2014/01/16/archaeologic_ian_williamson/.

ABC della merenda… opsss… del neofita appassionato!

…ovvero come arrivare ad usare un emulatore di retro computer al meglio e vivere felici!

di Ermanno Betori

In questa rivista abbiamo già scritto numerosi articoli di vario genere ma quasi tutti quelli tecnici danno per scontato una certa conoscenza più o meno profonda di linguaggi di programmazione, di come è costruito a livello hardware il computer e di come si usa. Ma se questo è vero per i vecchi appassionati, lo è di meno per le giovani leve che hanno solamente sentito per interposta persona cosa erano, come si usavano e cosa potevano offrire i vecchi computer. Per ovviare a questa situazione, inauguriamo con una serie di articoli, un mini corso che ha lo scopo ultimo di spiegare al neofita come usare in modo appropriato l'emulatore del computer preferito.

Per fare ciò il corso sarà idealmente diviso in tre sezioni..

1 - Spiegheremo come prima cosa il computer in esame, partendo dalla unità centrale, che ha avuto a livello hardware una sua evoluzione tramite l'utilizzo di vecchie e nuove periferiche le quali verranno poi integrate nei vari emulatori della macchina.

2- Presenteremo in dettaglio i principali emulatori della macchina, o che hanno specifiche particolarità.

3- Il metodo per interfacciare il vero computer con gli emulatori per creare-digitare-eseguire i programmi, scambiare i files - dump di dischi, ricreare i dischi, cassette per la macchina originale ecc…

Così speriamo che alla fine chiunque voglia cimentarsi sia in grado di avere il necessario background per usare al meglio gli emulatori, e che conosca in maniera più approfondita come la sua macchina sia "cresciuta" negli anni.

Auguro a tutti una buona lettura.

Il corso parte presentando uno dei tre nonnetti fine anni 70 che furono i primi ad avere una diffusione di massa.. il TI99/4A della Texas Instruments.

TI99/4A – ATTO I - Prima parte: I Computer

Nella linea di personal computer della Texas Instrument costruiti a fine anni 70, ci fu il primo computer a 16 bit rilasciato per le masse! Sebbene avesse l'interprete BASIC incorporato non pienamente efficace , i programmi scritti in linguaggio assembly erano di eccellenza. Di modelli la Texas ne creò quattro:

TI99/4 (anno 1978-79)

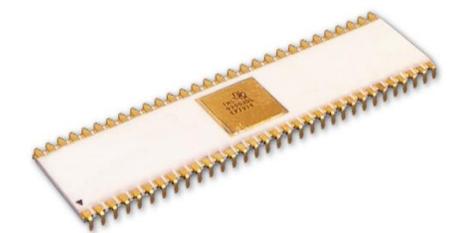
Nel giugno del 1979, la Texas Instruments commercializzò il TI99 / 4 personal computer, con un prezzo iniziale di $ 1500, che includeva un monitor. Ciò era dovuto al fatto che il modulatore, utilizzato per collegare il computer a un televisore, non superava gli standard americani di radio-frequenza. Particolarità di questa macchina era che aveva un programma di calcolo delle equazioni installato insieme al basic. Inoltre per avere una macchina completa di tutte le sue componenti un acquirente avrebbe dovuto avere molto spazio a disposizione, infatti le periferiche venivano inserite in una modalità in cascata che causava molti falsi contatti e faceva diventare il computer lungo come un trenino… vedi foto ☺.

Qui abbiamo la postazione ideale dell'epoca composta da consolle ti99/4,speech synthesizer (sintetizzatore vocale),stampante termica,espansione di memoria da 32k,espansione seriale/parallela,controller floppy drive,floppy drive da 90Kb,modem acustico da 300 baud. Furono prodotti oltre 20.000 di questi computer. Oggi sono un grande oggetto da collezione, e molto rari.

Ti99/4A (anno 1981-82)

Nel giugno del 1981 la Texas Instruments rilasciò il modello TI99/4A che in pratica fu una rielaborazione con migliorie del primo modello. La "A" proviene dal nuovo e migliorato processore video, il TMS9918A.

Questo modello aveva anche una tastiera molto migliorata. Il prezzo di rilascio per questo modello era di $ 525, senza monitor. Il modulatore finalmente aveva superato i rigidi requisiti FTC e finalmente era possibile collegare il computer alla TV di casa. (Si ricorda che agli inizi degli anni 80 la televisione a colori era un privilegio per pochi e che pochissimi televisori avevano altri ingressi video oltre alla normale presa RF, infatti all'epoca in Italia quasi tutti i computer usavano un modulatore che si agganciava di solito alla frequenza 36 UHF).

Come ho già scritto, uno dei miglioramenti del TI-99/4A rispetto al TI-99/4 fu la tastiera notevolmente migliorata. Comparando le foto notiamo che quella del TI-99/4A aveva 48 tasti invece di 41, caratteri maiuscoli e minuscoli e una maggiore solidità rispetto alla tastiera in gomma del TI-99/4.

Al contrario invece il ti99/4 aveva una CPU migliore come costruzione a livello tecnologico. Infatti era di ceramica con connettori in bagno di oro derivata da specifiche militari, rispetto a quella in plastica che fu usata per il ti99/4A

TI99/4A colore Beige (1983)

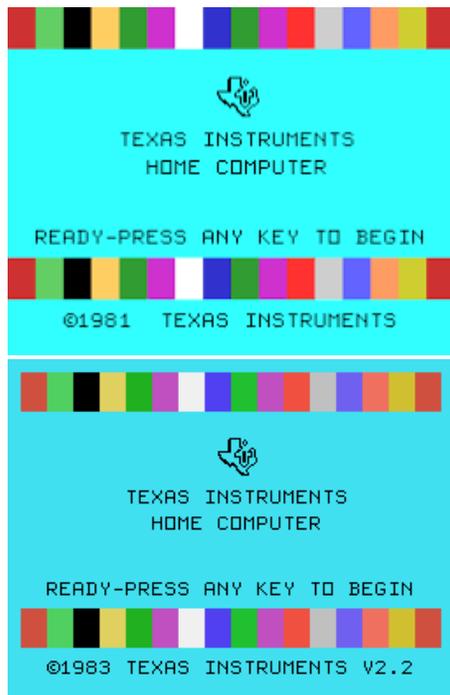
Nel giugno del 1983 venne rilasciata la versione in plastica "beige" della TI-99 / 4A. Fu creato con lo scopo di ridurre i costi di produzione e il colore fu scelto beige affinché si adattasse alla casa in modo più confortevole (stessa scelta verrà fatta anche negli anni successivi dai costruttori dei computer XT-AT).

In questa versione vennero fatte altre modifiche. Un nuovo alimentatore switching, la posizione dell'interruttore di alimentazione che viene spostato dalla parte anteriore a quella superiore.

Per il resto questo computer era lo stesso del modello nero e argento e poteva usare tutti gli stessi componenti aggiuntivi.

Una nota umoristica avveniva durante l'uso delle cartucce. Infatti la porta della cartuccia sulla console TI 99 era si posizionata in alto e facilmente raggiungibile, ma le prime cartucce furono prodotte in plastica nera, (per essere coeve con il design nero/argento) mentre le successive erano prodotte nel colore beige, con il risultato che si poteva avere un non felice accostamento estetico (vedi le figure sovrastanti). Una cosa bella del TI, al contrario della maggior parte dei computer del tempo, era che si potevano inserire e rimuovere le cartucce senza spegnere la console (quello sarà nel futuro denominato tecnologia hotswap). Un'eccezione a questa possibilità di uso era la cartuccia Mini Memory che conteneva una batteria interna.

TI99/4A Q.I.

Nell'agosto del 1983 la TI rilasciò il TI-99 / 4A QI che fu il suo ultimo modello che mise in commercio. Sebbene questo nome non fosse ufficiale per la console, era quello che veniva designato per la scheda madre. QI significa qualità migliorata. All'esterno della console non furono apportate modifiche visive e in effetti sono molto difficili da distinguere dai modelli beige. D'altra parte, furono apportate importanti modifiche ai componenti interni del QI specialmente sulla scheda madre.

Il più importante cambiamento che suscitò un mare di polemiche, era la modifica al firmware della scheda madre per bloccare le cartucce ROM senza licenza. Ciò fu fatto per impedire ad altre società terze di produrre cartucce per TI-99/4A senza avere il beneplacito della TI. Non tutte le console QI avevano questo "upgrade" che invece ritrovammo installato in tutte le ultime consolle in vendita sia che erano modelli 4A neri che beige. Infatti io comprai nel 1983 un TI99 nero argento con il firmware 2.2 che di fatto bloccava i giochi su cartuccia della atari tipo Pacman,Donkey Kong etc.. Per il neofita è facile determinare se si dispone del blocco o meno. Guarda la prima schermata (quella con le barre dei colori come in figura) che compare quando accendi la console, se mostra una data di copyright del 1981 non hai il blocco, ma se mostra una data di copyright del 1983 e il numero di versione v.2.2, allora è presente questo "nuovo" firmware. (Dopo anni scoprii come downgradare il firmware avendo i componenti giusti. Era di una facilità unica).

Questo alimentatore switching di "nuova concezione" (SX) rispetto al modello "tradizionale" (DX) fu montato nel computer dal 1983. Quando la TI uscì dal mercato nel 1984 mise in vendita a prezzi veramente fallimentari il TI99, e le macchine furono assemblate con le scorte di magazzino. Così in Italia vennero venduti per pochissimo tempo computer nero-argento con un mix di componenti destinati al beige e viceversa. Per la cronaca il nuovo modello mi si guastò, il tradizionale funziona sempre anche dopo 35 anni.

Queste motherboard sono rispettivamente quella del QI (alto) e quella standard (basso). Come già detto il QI doveva essere più competitivo a livello di costi per contrastare i vari concorrenti, infatti notiamo sulla standard 42 integrati contro 35. Fu il canto del cigno della TI.

Ringraziamenti

Si ringrazia Ron Reuter creatore del sito MainByte per le splendide foto e informazioni usate per la stesura di questo articolo. http://www.mainbyte.com/ti99/index.html

Altre macchine che rimasero a livello di prototipi furono il TI99/8 e il TI99/2.

il TI-99/8 doveva essere la risposta definitiva della TI ai critici del TI-99/4A e alle aziende che lottavano negli anni 80 per avere spazio nel mercato degli home-computer domestici. Sfortunatamente, il destino decise diversamente. Solo 150 prototipi sono stati costruiti prima che la TI si ritirasse dal mercato, utilizzando due versioni molto diverse della scheda madre. Circa metà delle macchine finite era di ciascun tipo di scheda madre, limitando ulteriormente l'utilità di alcuni dei dispositivi periferici

Come caratteristiche tecniche il TI99/8 era un miglioramento in quasi tutto, più ram, migliore basic, ecc.. mancando però l'upgrade fondamentale, cioè una vera e propria miglioria del processore grafico. Infatti il chip grafico che usarono era quasi identico come caratteristiche al quello montato sul TI99/4A.

Il Ti99/2 di contro doveva essere un computer a basso costo che doveva competere contro macchine tipo lo ZX Spectrum, ma la guerra dei prezzi che portò quasi al fallimento della TI e al suo ritiro dal mondo degli home computer fece rimanere un prototipo anche questa macchina.

Questa che vi abbiamo proposto è una presentazione basilare dei vari home computer della Texas Instruments serie TI99. Caratteristica interessante prima di addentrarvi nel mondo degli emulatori sarà farvi da cicerone per mostrarvi le molteplici periferiche che furono create per questi computer.

Anche se la produzione di questi computer si è conclusa nel 1984, possono essere ancora oggi utilizzati seriamente sia in campo educativo che in quello lavorativo, sempre con i dovuti limiti ovviamente!!!!

THEC64 Mini…

di Mattia Mottini, Edoardo Ullo e Marco Pistorio

In redazione ci siamo interrogati molto su come poter proporre una recensione del THEC64 Mini che fosse il piu' imparziale possibile ed al tempo stesso potesse fornire indicazioni valide a chi ancora e' indeciso se acquistarlo o meno. Abbiamo quindi concordato nel fornire il parere di 3 recensori, ognuno con il proprio bagaglio di esperienza e la sua visione dell'oggetto. Speriamo che questo esperimento possa incontrare il vostro favore e aiutarvi nel prendere una decisione.

Impressioni di Mattia Mottini

In questi giorni ha fatto la sua comparsa nei negozi, creando un boato nei vari gruppi dedicati, questa riproduzione in formato mini (scala 1:2) dell' home computer più venduto dell'era 8 bit che ancora oggi nutre uno stuolo di appassionati, non che di programmatori che, nonostante i 30 anni suonati della macchina, crea ancora software (per lo più giochi) dedicato.

L'uscita di questa nuova mini retro console (perché in fin dei conti è di una console che si tratta) ha spaccato non poco il popolo di appassionati, tra i puristi dell'hardware originale che al massimo si concedeno all'emulazione via PC considerandola una becera operazione marketing e chi invece ha accolto quest'oggettino a braccia aperte. Inutile dire che io faccio parte della seconda schiera, e gia dal 29 di settembre 2017 (giorno dell'apertura dei pre ordini) mi fiondai subito a fare l'ordinazione. Nel mentre i mesi passano, al games week di Milano conosco Paul Andrews e tocco con mano il primo prototipo di 64 mini facendomi salire l'hype alle stelle nell'attesa di poter stringere tra le mani la mia mini console per poterla testare approfonditamente.

Ma veniamo al punto, essendo io nato nel 1990 non ho potuto vivere l'era degli 8 bit in prima persona quando questa tecnologia era nel pieno della sua vita, e in casa mia ahimè l'hardware originale è assente. L'avvento di queste mini console (di marche ufficiali come ad esempio Nintendo, oppure di terze parti come il C64 mini che di commodore non ha nemmeno l'ombra) fù per me motivo di shopping compulsivo, ma nonostante tutto mancava sempre qualcosa. Quasi tutte le mini console (a meno di ricorrere all'hacking) hanno una lista di titoli che spaziano da capolavori a titoli sconosciuti molto ristretta e non ampliabile a meno di ricorrere ai metodi messi tra parentesi.

Ed è qui che esce uno dei lati positivi del C64 mini, ovvero lista di 64 titoli che possono o meno piacere e la possibilità di caricare via USB qualsiasi altra *rom* a nostro piacimento; feature non da poco e che toglie quel senso di castrazione presente sulle mini console di concorrenza e che al tempo stesso aumenta la longevità d'uso della mini console marchiata retrogames ltd. (e non Commodore, perché è bene ricordare che questa mini console è un tributo all'hardware originale e non un prodotto ufficiale della grande C).

Certo una volta arrivata a casa mia (il 29 marzo) la mini console si presentava in tutto il suo splendore, ma non senza qualche difettino di gioventù che la casa produttrice promette di risolvere man mano con futuri aggiornamenti firmware (al momento ne sono usciti 2), ma il potenziale c'è e secondo me si vede.

Fin dal primo giorno, nella curiosità generale mi misi subito a fare qualche test caricando le *rom* dei giochi che più mi piacciono riscontrando problemi solo con 2-3 titoli, probabilmente piu' per corruzione dei file che non per problemi dell'emulazione che di per sé trovo di notevole qualità. Abbinando una tastiera USB (la tastiera integrata è finta, e anche fosse stata vera servirebbero le dita di pollicino) con un hub USB riesco a tener collegati in porta 1 tastiera e joystick e in porta 2 una chiavetta usb che funge da drive 1541. Nel corso dei test ho provato sia giochi, che anche software generico come ad esempio il GEOS che parte regolarmente ma che ahimè non riconosce i movimenti del mouse a causa di uno di quelli che poco fa definivo difetti di gioventù, ovvero la mini console non riconosce il joystick se viene collegato alla porta 2, quindi al momento ho dovuto rinviare, anche se ho comunque potuto constatare che GEOS partiva senza particolari problemi.

Traendo le conclusioni finali direi che al momento la macchinetta assolve per buona parte il suo compito (con un tv LG 19" lcd non recentissimo non ho notato particolari problemi di lag) mi permette di avere vicino alla tv un oggettino che ad impatto visivo ricorda molto il Commodore 64, con un joystick plasticoso e dal sapore retrò che anche se non perfetto svolge comunque egregiamente la sua funzione, che mi permette se mi và di scrivere anche qualcosina in basic in quanto è presente nella lista software pre-caricata, che mi permette di aggiungere ulteriori titoli alla libreria che di per sé contiene gia 64 titoli (anche se al momento è ancora troppo macchinoso il caricamento di nuovo software soprattutto se questo è multi-disco) e mi permette di fare una scorpacciata di anni 80 (con tutti i limiti del caso ricordandomi sempre che in ogni caso siamo nel 2018).

Diciamo che se dovessi dargli un voto al momento gli darei un bel 7+ con provvisorietà dovuta al fatto che se Retrogames ltd. manterrà le sue promesse e continuerà a lavorare per correggere gli errori potrebbe portarci ad avere la retro console definitiva, che in pochissimi minuti ci permetterebbe di goderci il nostro titolo preferito: giusto il tempo di collegare il cavo HDMI, il cavo USB, e go siamo pronti a giocare. Trovo che sia un

buon compromesso per chi non ha l'hardware originale e non ha voglia di cimentarsi con emulatori per pc o raspberry vari, o per chi l'hardware originale vuole conservarlo e/o lo tiene ritirato al sicuro per mancanza di spazio in casa o per paura di rovinarlo.

Impressioni di Edoardo Ullo

È indubbio che il THEC64 mini abbia fatto parlare di sé gli appassionati. Soprattutto quelli di vecchia data. Quest'ultimi si sono divisi tra chi ha criticato aspramente questa riproposizione realizzata da Retro Games LTD ed altri invece che l'hanno apprezzata.

Parto dal principio che al di là dei ricordi e delle emozioni che possa rievocare, si tratta di un'operazione nostalgia e commerciale alla quale, però, guardo in modo piuttosto favorevole.

Il perché è presto detto: mi pare comunque carino ricordare, o rievocare, un computer glorioso come il Commodore 64. Detto questo, comincio subito a parlare dei pregi e dei difetti che personalmente ho riscontrato nelle mie ore di prova. E parto dai tanti pregi. In primis il software. È ottimo, funzionale, non crea problemi ed è alla portata di tutti. Le opzioni supplementari non si discostano troppo da quanto proposto da alcune altre riproposizioni, leggasi i Nes e SNES Classic mini. Pertanto è possibile scegliere il formato (PAL o NTSC), il formato in 4:3 o altre opzioni quali filtri grafici che rendono più o meno vintage l'impatto visivo. Il menu di navigazione (accompagnato da una musica scritta per l'occasione da Matt Grey) è immediato e permette subito di giocare ai 63 giochi (più il Basic che però non è un gioco). Altra nota a favore: scelto un gioco, questo si carica in modo istantaneo, senza cioè aspettare i lunghi tempi di caricamento di cassette o dischetti. Del resto siamo nel 2018 e qualche passetto avanti rispetto al 1982 lo si è comunque fatto e questo si concretizza con un attacco HDMI per l'attacco ai moderni televisori. C'è anche una comoda funzione per salvare la partita e riprenderla quando si vuole ma c'è la grande possibilità di caricare, anche se per ora in modo macchinoso, le immagini (.D64) di altri giochi per potersi cimentare in altri giochi e/o programmi o demo.

Diversamente potrei tranquillamente parlare di una mezza delusione perché i titoli presenti sono altalenanti. Intendiamoci, molti hanno davvero scritto la storia e pensiamo alla presenza di Boulder Dash, o all'incredibile Creatures, ai capolavori artistici che rispondono al nome di Cybernoid 1 e 2 dove la palette sembra miracolata e la musica di Jeroen Tell riecheggia in modo imperioso. Tra altri titoli storici non possiamo non menzionare Armylate (e la sua bellezza lineare), Hawkeye (ed i suoi livelli di parallasse), o i divertenti California Games, World Games, gli sportivi Summer Games e Winter Games, i frenetici Speedball nonché i due Impossible Mission. Poi tantissime vecchie glorie che però ci fanno pesare l'assenza di altri classici quali i Turrican, i The Last Ninja e così via. Questione di diritti.

È un grosso punto a suo favore, quindi, la possibilità di caricare altri giochi benché la sua procedura sia, al momento complicata soprattutto se i titoli desiderati risiedono su più dischi. Per fortuna gli sviluppatori stanno pensando a questo ed il prossimo aggiornamento dovrebbe facilitare il tutto.

È possibile, quindi, aggiornare il THEC64 Mini con USB dopo aver scaricato il firmware direttamente dal sito ufficiale.

C'è anche la presenza di una tastiera virtuale, scomoda ma alla fine funzionale che comunque può essere ovviata connettendo tramite porta USB una moderna tastiera. Sarà possibile anche programmare col Basic e fare quelle operazioni necessarie per caricare software esterno.

I puristi potranno obiettare sul fatto che la tastiera sia solo figurativa. Questa infatti non funziona e non ha con sé neppure i comandi Basic riportati nei tasti dell'originale. Ma il vero punto debole è il joystick. Nella forma è simile al Copetition Pro ma è senza microswitch ed offre alcuni pulsanti in più. Si notano i due Fire classici laterali in avanti, affiancati da due triangolari posti quasi alla base della levetta e quattro pulsanti tondi nella parte posteriore. Uno di questi (quello a destra) serve per aprire i menu di navigazione per le varie opzioni supplementari. Ma il problema non sono i pulsanti. La sensazione di fragilità è tale che, memore delle esperienze con joystick simili negli anni '80 e '90, gioco in modo timido perché sforzandolo come potrei e dovrei per giocare al meglio, sono certo che lo romperei in pochissimi istanti. Un vero peccato perché questo non permette di divertirsi al meglio. Inoltre diversi joypad moderni non sono supportati. In teoria sarebbe poco male ma a questo punto, vista la pochezza del joystick, vorrei un'alternativa più solida.

Complessivamente, il THEC64 Mini mi è piaciuto molto per i motivi che ho elencato. Non è perfetto ma è aggiornabile e è la line-up (alla quale di base darei comunque un buon 7 perché include almeno una dozzina di titoli di livello assoluto) è espandibile. Il software (aggiornabile facilmente via USB) inoltre funziona molto bene e non c'è bisogno di armeggiare con varie opzioni per giocare immediatamente. Peccato davvero per il joystick, davvero troppo fragile e neppure troppo preciso. E per la tastiera non funzionante pazienza: del resto non mi ci vedrei a scrivere linee di programma con una tastierina che è più piccola della metà del Commodore 64 reale. Ultimo ma non ultimo... avrebbero potuto metterlo un trasformatore 5V ad 1A. In molti avremmo pagato volentieri 5 euro in più.

Impressioni di Marco Pistorio

Cosa ne penso del "The C64 mini"?

Che l'unico, vero problema di questo prodotto è che ricorda troppo da vicino quel "mostro sacro" caro a tanti di noi che è il Commodore 64. Scherzo, voglio essere obiettivo.

Come in tutte le cose di questo mondo, c'è del buono e del cattivo anche nel "The C64 mini".

E' molto "user-friendly", intuitivo nell'utilizzo e nel collegamento, adeguato quindi alla fruizione sia da parte dei vecchi utenti che, soprattutto, da parte dei nuovi e nuovissimi utenti.

Il parco dei giochi "on-board", seppur limitato, contiene titoli qualitativamente buoni (insieme ad altri meno buoni), ma si tratta in tutti i casi di titoli regolarmente licenziati.

L'aspetto ricorda certamente molto il Commodore 64, aspetto importante perché crea una esperienza durante il suo utilizzo ben più intensa di quella che si prova di solito con

l'uso dei classici emulatori su PC. Manca il logo "Commodore", è vero. Mentre per i titoli dei giochi e per le ROMS CBM originali la società Retro Games ltd ha ottenuto le relative licenze, per quanto concerne il logo non è stato compiuto l'analogo sforzo per poterlo sfruttare nel rispetto della legislazione vigente.

Il motivo è presto detto. La storia del marchio "Commodore" a partire dal fallimento della Commodore Business Machines, con i vari soggetti che lo hanno acquisito per poi cederlo ancora, è una corsa ad ostacoli comprensiva di incognite nella quale la suddetta Retro Games ltd avrà deciso di non cimentarsi.

La tastiera, benchè visivamente molto simile a quella originale, è non funzionante.

E' vero che sarebbe possibile collegarne una funzionante ad una delle porte USB presenti (se non si fosse disposti ad adoperare quella "virtuale" su schermo TV), ma è anche vero che se la tastiera fosse stata perfettamente funzionante dal punto di vista meccanico, probabilmente sarebbe stata anche poco funzionale dal momento che l'intera tastiera rientra, grossomodo, nel palmo di una mano.

Speriamo che questo problema possa essere risolto con l'arrivo del "the c64" in scala 1:1 rispetto al vecchio "commie", che Retro Games ltd prevede di commercializzare entro l'anno.

Il joystick è funzionante e visivamente abbastanza accattivante ma qualitativamente non all'altezza.

Alla pagina https://thec64.com/accessories/ è scritto però che a breve termine ci saranno una serie di nuovi accessori per il "the c64", ed è probabile che verranno offerti prodotti migliori del joystick attualmente previsto.

Ritengo comunque problemi abbastanza seri sia la tastiera, che in questa versione non è funzionante, sia il joystick non adeguato.

Insieme ai titoli già presenti sul "the c64 mini" esiste la possibilità di caricarne dei nuovi.

Purtroppo il metodo è poco funzionale, in particolare se si ha intenzione di giocare a giochi che richiedono il caricamento di più di un floppy disk.

Inoltre, almeno per il momento, non è possibile usare immagini di cassette né di cartucce.Risulta spesso difficile poter giocare ai titoli che prevedono l'inserimento del joystick su porta 1, mentre invece quelli che prevedono l'uso del joystick in porta 2 funzionano regolarmente.

Questi problemi però potrebbero essere corretti con un aggiornamento del "firmware" del "the c64 mini", e mi auguro quindi che saranno presto superati.

Un sunto degli aspetti positivi del "the c64 mini"?

E' molto simile, visivamente almeno, al Commodore 64, tempi di caricamento dei giochi prossimi a zero, collegamento immediato al TV di casa, intuitivo nell'utilizzo, 64 titoli presenti di base, uscita video ed audio di qualità HDMI, ideale per fare conoscere soprattutto alle nuove generazioni il retrogaming (ed in particolare il retrogaming legato al Commodore 64), e con il supporto di una società che ne cura la distribuzione, l'aggiornamento e lo sviluppo in generale, ad oltre vent'anni dalla scomparsa dal mercato della Commodore Business Machine, è bene tenere presente anche questo aspetto.

Il "the c64 mini" non è perfetto, c'è ancora parecchio da fare.

Tuttavia è una novità che merita di essere accolta, esattamente come lo merita qualsiasi progetto attuale, che permetta esperienze di retrogaming attirando users vecchi e nuovi.

E se nella redazione di "RetroMagazine" ci siamo fatti in quattro per il "the c64 mini" (è proprio il caso di dirlo) probabilmente lo merita!

Ciao a tutti :)

Conclusioni

Se vi aspettate delle conclusioni, rimarrete delusi, ognuno e' libero di trarre le proprie e scegliere o meno se acquistare il THEC64 Mini; da parte nostra speriamo solo di aver comunicato le nostre impressioni nel modo piu' corretto possibile elencando in modo pragmatico ed imparziale i pregi ed i difetti di questo oggetto.

Aggiungo inoltre che ci e' stato richiesto piu' volte di recensire oltre al Software (giochi e quant'altro) anche l'Hardware e questo esperimento ci fara' capire se l'argomento e' di vostro interesse. Fateci sapere!

Console 8bit: **Atari 5200**

di Starfox Mulder

Atari non fu solo console ma nel vasto mercato dell'intrattenimento elettronico si dedicò anche ad una sua linea di computer ad 8-bit, in concorrenza diretta con Apple II, TRS-80 e Commodore PET. Candy e Colleen, rispettivamente i nomi femminili associati all'Atari 400 ed al 800, montavano una CPU Mos 6502, coprocessori ANTIC e GTIA (per il video) e POKEY (per il sonoro). Quando ormai sul fronte console erano arrivate agguerrite le avversarie di marchio Mattel e Coleco, Atari pensò (bene) di "consolizzare" l'hardware della sua linea di home computer ad 8bit ma pensò anche (male) di non rendere i due sistemi direttamente compatibili. L'Atari 5200, nome in codice SuperSystem, aveva sì il vantaggio di permettere una facile conversione ma di fatto quella andava fatta e non è che la cosa si risolvesse in una giornata ma bensì in un ben lungo processo di produzione-distribuzione in cui i costi necessitavano di essere coperti con vendite che non arrivarono mai.

Le premesse erano comunque valide? Difficile a dirsi. La console presentava inizialmente ben 4 porte joystick, i nuovi controller strizzavano l'occhio a quelli delle competitor (analogico + tastierino numerico) ma erano decisamente imprecisi ed il parco titoli era votato completamente al settore arcade, presentando però solo due titoli esclusivi: Countermeasure e Meteorites.

Tutti gli altri? Conversioni "pimpate" dei giochi già apparsi su Atari 400/800. Belli, quasi in tutto superiori alle controparti per 2600, eppure non sufficienti a spostare il mercato.

Un'altra caratteristica particolare del SuperSystem fu l'utilizzo del collegamento "switchbox", un particolare sistema per trasportare sullo stesso cavo sia l'alimentazione che il segnale video, collegato quindi a sua volta sia con la corrente di casa che con la TV.

Il più grosso auto-sabotaggio che Atari si fece però fu legato al mancato adattatore per titoli VCS. Un apparecchio simile era stato prodotto sia per Intellivision che per Colecovision e vendette molto bene in quest'ultimo caso, al punto che non solo divenne per Atari un motivo di freno alle vendite, ma se ne fecero in seguito vanto per poter lanciare la successiva console: il 7800 (già di default retro-compatibile con il 2600).

La seconda versione del 5200 perse due porte joystick e la switchbox, limitandosi a guadagnare la compatibilità con il ritardatario adattatore per cartucce VCS, ma come potete intuire: ormai era tardi.

La console, per quanto ingombrante, era di fatto dotata di un hardware all'avanguardia, superiore alla rivale Coleco e con un buon numero di giochi degni di essere posseduti, ma le scelte sbagliate che Atari fece in merito alla retro-compatibilità ed il suo esser stata commercializzata a ridosso della crisi dei videogiochi fecero sì che per il pubblico non apparisse in alcun modo desiderabile o preferibile alle avversarie, se non al massimo per il nome che portava

Alla prossima console!

CARATTERISTICHE TECNICHE

Produttore	Atari
Tipo	Home Video Game Console
Generazione	Seconda
In vendita	Novembre 1982
Dismissione	Maggio 1984
Supporto	Cartuccia
Unita' vendute	1 milione circa

FONTE: HTTPS://EN.WIKIPEDIA.ORG/WIKI/ATARI_5200

SENSIBLE SOCCER

Che bel panorama...

Classica visuale dall'alto che richiama subito alla mente docissimi ricordi.

Tattiche e moduli

Prima di ogni incontro è possibile scegliere i titolari, il modulo e sbirciare la tattica dell'avversario.

Immagine pre-partita

Una bella schermata prima del fischio d'inizio ci mostrava in anticipo le divise delle squadre.

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

Il calcio è sicuramente lo sport più popolare ed amato nel mondo.

LONGEVITA'

90%

La community di SWOS e' tuttora molto attiva online!

Un colosso tra i giganti... ecco Sensible Soccer!

Sensible Software - Anno 1992 - Piattaforma Sega Megadrive

Il calcio è sicuramente lo sport più popolare ed amato nel mondo e proprio per questo le maggiori aziende produttrici di videogame hanno tentato in questi anni di proporre, con alterne fortune, i propri titoli di simulazione calcistica.
Oggi il panorama mondiale è dominato da due colossi, sto parlando di Pes della Konami e di Fifa della Elettronic Arts, che, non senza esclusione di colpi, si stanno dividendo ogni anno il podio di migliore gioco di calcio.

Ma se state leggendo questa rivista siete anche voi amanti del retrogame, perciò vi invito a fare un piccolo esperimento. Chiudete per un momento gli occhi e, seppure non disponiamo della mitica DeLorean di Marty McFly e di Doc per andare nel passato, proviamo lo stesso a fare un salto indietro nei nostri ricordi e passiamo in rassegna tutte le immagini di giochi di calcio che hanno riempito le nostre giornate di adolescenti. Sono sicuro che tra i vari titoli ce ne è uno che al suo ricordo vi strapperà un dolce sorriso: sto parlando naturalmente di Sensible Soccer.
Nato dalla mente di Jon Hare, che si era già guadagnato fama ed onore con Microprose Soccer per Commodore 64, e distribuito dalla Sensible Software, Sensible Soccer fa la sua comparsa nel 1992 su piattaforma Amiga ed è subito un gran successo. I videogiocatori si innamorano immediatamente di quei paffuti e colorati calciatori che si muovono in modo frenetico sul campo di calcio dandosi battaglia tra passaggi veloci, tiri ad effetto, scivolate e colpi di testa volanti. Così, solo dopo due anni, viene rilasciato Sensible World of Soccer (per i fan del genere soltanto Swos) che arricchisce il collaudato gameplay di numerose squadre e soprattutto di una modalità carriera che rimarrà storica. Inoltre il titolo viene prodotto per le console Super Nintendo e Megadrive. Ed è proprio di quest'ultima conversione che voglio parlarvi, un'autentica perla per ammissione del suo stesso creatore Jon Hare.

Nella versione per casa Sega, infatti, sono riusciti a mantenere la stessa struttura del gameplay originale con la classica visuale dall'alto a volo di uccello e con le stesse dinamiche di gioco della prima versione. Inoltre gli sviluppatori hanno confermato la stessa possibilità di organizzare amichevoli, coppe, tornei e campionati scegliendo tra le tante squadre nazionali e di club. Non manca nemmeno un bellissimo editor che da la possibilità di aggiornare squadre, divise e rose (che tra l'altro avevano già i nomi originali dei giocatori) e quindi di mantenere il gioco praticamente sempre attuale. Inoltre in questa versione è stata prevista una modifica che a mio avviso mancava e cioè l'abbinamento ai diversi tasti del gamepad del tiro, del passaggio semplice e di un pallonetto che può essere sfruttato per passaggi volanti o per autentiche sassate da lanciare verso il portiere avversario. Tutte possibilità che, nella versione originale, erano affidate ad un solo tasto.

So bene che la community di Swos (tra l'altro oggi ancora molto attiva online sia in Italia che all'estero) potrebbe non essere d'accordo su quanto detto finora e ribattere che la fisica del pallone nella versione Amiga è tutt'altra cosa, ma qui non voglio assolutamente dimostrare che una è migliore dell'altra, ma solo affermare che, una volta tanto, una conversione per console è perfettamente riuscita.

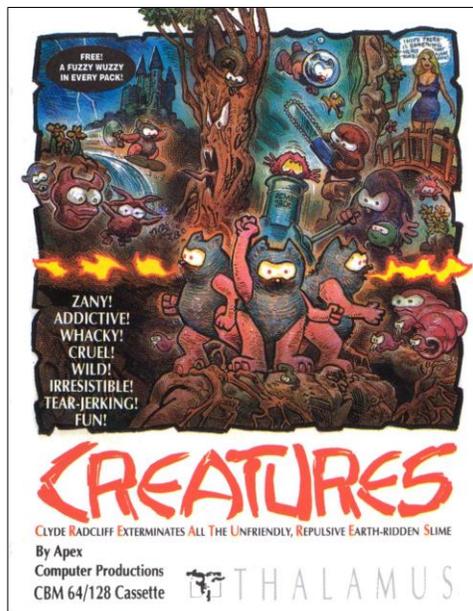
Tra l'altro la community di Swos non è l'unica attiva perché recentemente è nato un gruppo di appassionati di retrogames che, sfruttando la funzionalità del netplay del front end di Retroarch, ha deciso di ritrovarsi online per giocare insieme le vecchie glorie del retrogame in modalità multiplayer fino a 4 giocatori contemporaneamente. Trovate questi eterni adolescenti su facebook al link https://www.facebook.com/groups/retromultiplayer/ e anche sul server discord https://discord.gg/rAkfSrD.

Inutile dire che Sensible Soccer in versione Megadrive è uno dei titoli da loro più giocati e amati.

di Querino Ialongo

CREATURES

Creatures

Thalamus - Anno 1990 - Piattaforme: C64, Amiga, Atari ST

La stanza delle torture

I demoni sorridono, i Fuzzy no!

GIUDIZIO SUL GIOCO

GIOCABILITA'

90%

Il timing delle azioni è pressochè al nanosecondo, bisogna addentrarsi con adrenalina e mano ferma.

LONGEVITA'

95%

E' presente da poche settimane anche sul neonato TheC64Mini, gli anni 90 sono rinati!

Ma stiamo scherzando?

Un demonio di mostriciattolo, in un buio angolo di una caverna angusta, munito di motosega e strumenti della famosa ACME, sta letteralmente tagliando a metà Chip, la mia amica tartaruga?

Ecco a voi Creatures, una buffa storia di adorabili esseri chiamati "Blotiani" che dal noioso pianeta "Blot" ai confini della Via Lattea partono alla ricerca di un pianeta più divertente. Nella nuova colonia decideranno di chiamarsi "Fuzzy Wuzzies".

Purtroppo però durante il viaggio spaziale, dopo una collisione con un asteroide, precipitano in una sconosciuta isoletta dell'Oceano Pacifico. Qui cominciano i guai, si beh l'isola è bellissima e incontaminata... ma anche piena zeppa di demoni.

I Fuzzy sono pelosi, simpatici, giocosi, rumorosi, ogni momento è giusto per organizzare una festa! I demoni autoctoni però, dopo aver a lungo sopportato questi ospiti poco graditi, stanchi del loro frastuono, un bel giorno inscenano un irresistibile party, li invitano e li catturano tutti quanti.

Che fine faranno i poveri Fuzzy?

I demoni li portano nelle stanze delle torture per eliminarli ad uno ad uno spietatamente.

La fortuna ha voluto che non tutti i Fuzzy venissero catturati, ecco che nasce involontariamente l'eroe del gioco, cioè il protagonista: Clyde Radcliff.

Infatti la mattina dopo la finta festa, Clyde si sveglia in mezzo alla vegetazione con un gran mal di testa e, poveretto, si ritrova in una situazione agghiacciante. Amici rapiti, demoni sanguinari tutti attorno, armato unicamente del suo alito infuocato dovrà sconfiggere tutti i demoni e salvare gli altri Fuzzy rapiti che ritroverà in alcune scene incatenati, in altre mangiucchiati, tagliati, smembrati, etc…

Da qui nasce l'acronimo CREATURES: Clyde Radcliffe Exterminates All The Unfriendly Repulsive Earth-ridden Slime, in parole povere Clyde, da solo, dovrà sterminare tutta la feccia di demoni.

Questa "medaglia d'oro" insignita dalla rivista "ZZap!", anche intitolata "Torture trouble", con punteggio 96% è stata pubblicata quasi un terzo di secolo fa, nel 1990 dalla Thalamus.

Non è prettamente un genere platform a scorrimento orizzontale.

Difatti alla fine di ognuno dei tre livelli, ciascuno composto da tre sottolivelli, proprio questo terzo viene rappresentato da un'unica schermata, la famosa stanza delle torture, tipo puzzle, composto da un enigma da risolvere rapidamente per salvare l'amico Chip dalle atroci torture. In totale tre stanze delle torture! In più ad ogni fine livello una strega donerà una pozione magica a Clyde in cambio di mostriciattoli raccolti nel percorso. La pozione aggiungerà nuove modalità di fuoco.

I fratelli Rowlands elaboreranno alla perfezione, uno l'audio e la grafica, l'altro la programmazione. In questo capolavoro di musiche e grafiche si potrà dire che l'uno diverrà la motivazione dell'altro, basta leggere l'avvincente diario della creazione del gioco (vedi Approfondimenti). Ebbene, per numerosi e solidi aspetti, la Thalamus e i Rowlands, a breve, avrebbero regalato arte pura al mondo.

Elenchiamo alcuni di questi aspetti per apprezzare appieno questo gioco e sopratutto per poterlo annoverare tra i migliori giochi "allo stato dell'arte":

Il chip video è spremuto per dare un rendering con scrolling assolutamente fluido, tantissime animazioni, numerosi demoni e mostriciattoli, tutti in movimento o prossimi a muoversi se risvegliati da qualche azione.

Quasi sempre ogni demone è dotato di un carattere differente dagli altri. Il nostro Clyde sbatte le palpebre, respira sott'acqua dentro una sorta di boccia di vetro per pesci rossi ed emette le bollicine, vola sulla scopa, salta, spara fuoco in numerose modalità, danza, passeggia, se lasciato immobile ci avvisa dello

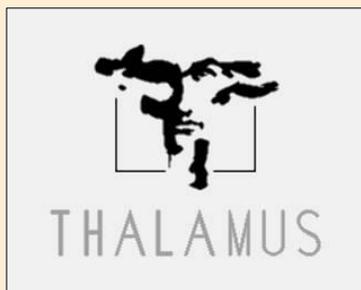
scorrere del tempo, se ci avviciniamo a schermate che contengono acqua corrente sentiremo aumentare il volume del rumore dell'acqua, se ci allontaniamo il volume diminuirà.

E' tanta la ricchezza di particolarità: la strega che compare a fine livello è stata creata da una ispirazione del famoso Playboy e misura 36-24-36 (misure ovviamente inglesi).

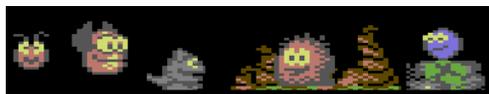
Nella stanza delle torture dove Chip morirà tagliato dalla sega da banco (o scossa elettrica?) sono state programmate ben 122

CREATURES

Clyde Radcliffe Exterminates All The Unfriendly Repulsive Earth-ridden Slime, in parole povere Clyde, da solo, dovrà sterminare tutta la feccia di demoni..

CURIOSITA'

Nel famoso sito http://blowthecartridge.com si troveranno delle divertenti vignette umoristiche sui Fuzzy Wuzzies

In questi anni le musiche dei Rowlands sono state amate veramente da tantissimi giocatori, tanto che alcuni, dotati di spiccata creatività audiofila, hanno composto musiche e remix sulle orme dei due titoli, basta cercare "Creatures" nel sito dei remix per C64 http://remix.kwed.org

Chissà se qualcuno assaporando il magico motivo di Jouni Vepsäläinen ricorderà il titolo "The Fantastic Oceans - Sunshine" nel film di Bud Spencer e Terence Hill "Chi trova un amico trova un tesoro" dove anch'essi erano naufraghi in una sperduta isoletta incontaminata?

APPROFONDIMENTO

Chi gradirà approfondimenti della saga Creatures potrà leggere questo simpatico diario dei fratelli Rowlands:

http://www.gamestone.co.uk/zzap_world64/apex_creatures.php

animazioni e 32 sprites. Nell'introduzione di ciascun sottolivello viene fatto un rapido scrolling del percorso che dovremo completare.

L'introduzione è composta da due demoni che vengono schiacciati dalla scritta Creatures mentre i Fuzzy danzano a tempo. Ogni scrolling di sottolivello ha suddivisioni e caratteristiche differenti per ciascuna sezione, basti pensare al sensazionale bosco (ed anche al cimitero) stile "Ghost and goblins" che scorrono nello sfondo con una fluidità ineccepibile.

Una fantastica metodica di fastflickering, miniaturizzata in alcuni sprite, supera il limite dei canonici 16 colori. Sebbene il timing sia gestito con ampie deviazioni tra i vari personaggi della schermata, è pressochè trascurabile ogni difetto di proiezione degli elementi. Infine se si riesce a completare il gioco, c'è un allegro finale!

I panorami includono grotte, caverne, sopra e sotto terra, per aria e anche in acqua con presenza di cascate, luci, zombie, fiaccole, gufi, alberi animati, pesci, zucche rotanti, uccellacci kamikaze, fantasmi, pipistrelli, fuochi fatui, mongolfiere bombardanti, rapidi granchi, meduse sorridenti, e... si, tanti... ma tanti... ma tanti odiosissimi sorrisi stampati a fuoco sulla faccia di quasi tutti i demoni tranne il nostro Fuzzy e ovviamente Chip, poveretto, come dargli torto, dato che sarà tagliato a metà?

Bug: certo che ci sono bug anche in questo gioco, ammettendo che Clyde è l'unico Fuzzy a non essere stato rapito, come si spiega la presenza della sua compagna in lacrime sul coraggioso defunto che non ha superato il livello? Che ci fa una topmodel dai lunghi capelli biondi con tubino nero attillato in una foresta popolata da demoni? Chissà perchè non ci balenavano queste domande alla tenera età in cui ci giocavamo? Inoltre Chip nel porting Amiga, sotto la motosega sembra un Fuzzy, mentre nel C64 sembra una tartaruga! Mumble mumble! Perchè alcuni demoni colpiti e arrostiti da Clyde continuano a mantenere un immarcescibile sorriso stampato sulla faccia, fino alla fine, stile film "la morte ti fa bella"?

Perchè i demoni non hanno socializzato con la strega o almeno eliminato tutti gli animaletti utilizzati come merce di scambio per la pozione destinata a Clyde?

Eh si, gli anni 90, che magia, che pace, quante inutili e maliziose e schizzinose domande mi pongo oggi, ripensandoci... non ci vedo nulla di male in una strega che possa provare simpatia per dei festosi e pacioccosi alieni! In fondo quei demoni non le sono mai piaciuti, castelli enormi circondati da infiniti zombie, uccellacci, fantasmi e .. dentro?

Mai neppure una piccolissima festa?

Poi, ripensandoci bene, i prodotti marchiati ACME non sono mai piaciuti neppure a "Willy il coyote".

Sono stati eseguiti porting di Creatures sia su Amiga che Atari ST. Sono state mosse numerose critiche riguardo il guadagno dei colori in Amiga a scapito di alcuni problemi: sovrapposizioni audio durante il gioco, rallentamenti, colori troppo saturi e grafiche meno accattivanti nei vari soggetti.

Rammentiamo comunque che stiamo parlando di gusti e sappiamo che questi sono l'essenza della personalità. Non siamo qui a criticare il porting di un capolavoro, cadremmo nel ridicolo, poichè parlando di preferenze, ognuno difenderebbe con orgoglio l'amore nato per questa saga. L'unica cosa sicura è che Thalamus ci ha resi tutti fratelli in questa passione di un capolavoro della storia videolutica.

Il business della Thalamus: insieme ai titoli Creatures 1 e 2 e numerosi altri, la ditta ha venduto tante di quelle cassette e floppy che nel corso degli anni ha seminato e coltivato in noi la latenza di una fortissima nostalgia. Una nostalgia così potente che da alcuni mesi ha fatto rinascere attraverso nuove vesti, cioè il moderno trend del crowdfunding, la "Thalamus Digital" e a furor di popolo sta già riproducendo alcune delle sue hit.

Amici miei, spero di non avervi annoiati con questa recensione, spero sopratutto che la passione di Creatures possa far nascere un fan club sui social e perchè no, un clone remake amatoriale tipo "Critters 2" per divulgare il più possibile questo piccolo capolavoro.

Questo primo titolo di Creatures, da Aprile 2018, è presente con regolare licenza anche nel neonato TheC64 Mini.

Vi invito tutti quanti a giocarci, amarlo quanto l'ho amato io, salvare Chip e tutti i Fuzzy Wuzzies e gustare il romantico e festoso finale.

Vi aspetto per un'altra recensione nella quale parlerò del secondo titolo cioè la costruzione di un secondo capolavoro che ha mantenuto l'anima del primo senza copiare tutto quanto, anzi, cercando di apportare migliorie ed innovazione, incluse più stanze delle torture!

di Michele Ugolini

Berzerk

Atari - Anno 1983 - Piattaforma Atari 5200

BERZERK

Effetti Sonori incredibili 1…

"Intruder alert! Intruder alert!" - all'apparizione di Evil Otto.

"The humanoid must not escape" – Uscendo da uno stage dopo aver ucciso tutti i robots.

Effetti Sonori incredibili 2…

"Chicken, fight like a robot" – Uscendo da uno stage senza aver ucciso tutti i robots.

"Got the humanoid, got the intruder!" - Alla morte del giocatore.

GIUDIZIO SUL GIOCO

GIOCABILITA'

85%

L'eccellente giocabilità del titolo Arcade si scontra contro l'unico difetto di questa versione: il controller del 5200. Nonostante l'imprecisione dello stesso se ne esce comunque decentemente ed il divertimento non perde di intensità.

LONGEVITA'

85%

Stage casuali, nemici posizionati a random e undici varianti di gioco rendono l'esperienza sempre varia ed appagante.

Lo so, Berzerk uscì per tutte le piattaforma Atari dell'epoca ma trattandosi del mese SuperSystem ho preso al volo l'occasione per parlare di uno dei miei Arcade preferiti di sempre.

Seguendo la politica dell'epoca il gioco non ha fine e si può solo avanzare all'inseguimento del record perfetto ma nel farlo ci verrà dato, stavolta, un bel pacchetto di ambientazione a cui ispirarci.

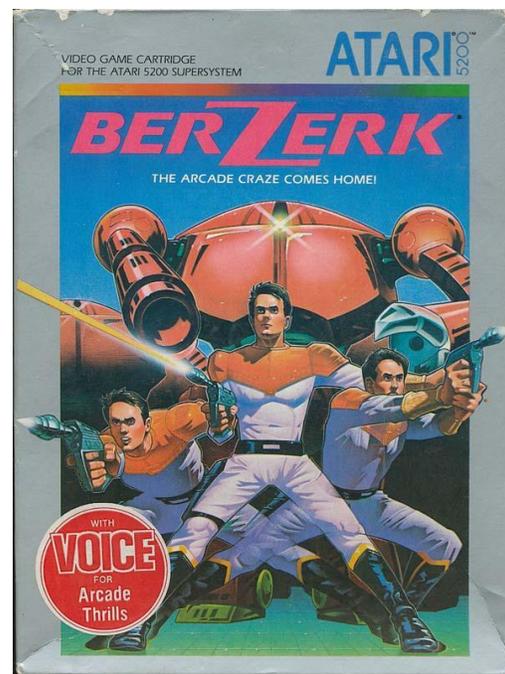
Alain McNeil, il programmatore che ideò il gioco per Stern (versione Arcade) era un fan della saga fantascientifica Berserker, di Fred Saberhagen. Nel ciclo di romanzi si narra di macchine costruite millenni prima dell'avvento della razza umana che nel tempo han sviluppato una loro intelligenza artificiale ed ora sono giunte fino a noi con il preciso scopo di sterminarci.

Il protagonista del gioco (noi) si trova a visitare un pianeta apparentemente disabitato quando la sua nave spaziale viene fatta saltare in aria ed i suoi compagni trucidati dagli Auto-Mazeons, i robot di qui sopra comandati da Evil Otto, un inquietante Smile saltellante che ci darà la caccia attraverso i vari stage finché non ci avrà acchiappato ed eliminato. Ma andiamo per gradi.

Il primo stage servirà a farci ambientare coi comandi, presentando un proto-tutorial, strano a trovarsi in un gioco arcade. I nemici sono fermi e non ci spareranno, sarà quindi facile per noi apprendere dei movimenti del nostro alter-ego, gestibili tramite il joystick, e dello sparo che potremo dirigere nelle medesime otto direzioni in cui ci staremo dirigendo in quel momento. Tenendo premuto lo sparo e spostando il joystick vedremo il nostro protagonista stare fermo e cambiare solo direzione di sparo. Per quanto il nemico non attacchi ci sarà comunque possibile morire toccando le pareti elettrificate dei labirinti, finendo troppo vicini alle esplosioni dei nemici colpiti o risultando abbastanza lenti nel nostro far piazza pulita di tutto e tutti, portando alla conseguente apparizione del temuto Evil Otto. Il boss saltellante, che prese il nome da un collega di lavoro particolarmente fastidioso del nostro Alain, compare dopo qualche decina di secondi di permanenza in uno stage e ci inseguirà passando oltre i muri come se nulla fosse e puntando direttamente a noi. Non potremo ucciderlo, solo sfuggirgli temporaneamente passando allo stage successivo, il che ci metterà in un costante stato di ansia da minaccia infinita, roba che Resident Evil 3 non ha proprio inventato nulla.

Giunti al secondo livello comunque anche gli auto-mazeons diverranno belli agguerriti sparandoci addossi da ogni direzione e dirigendosi verso di noi. Fortuna vuole che gli avversari non siano proprio dei geni di tattica e spesso finiranno con lo schiantarsi da soli contro i muri elettrificati, elargendoci punti e campo libero.

La pazienza ed il giusto tempismo sono il segreto per trionfare in Berzerk ed è proprio un caso palese di quanto per diventare sempre più bravi si debba giocare ed ancora giocare finché non si sarà diventati davvero competitivi. La versione per Atari 5200 è davvero un'ottima conversione del titolo arcade. Rispetto alla precedente per 2600 stavolta gli stage sono della grandezza giusta, la sintesi vocale è stata mantenuta (i nemici ci aggrediranno verbalmente, cosa che aiuta ad aumentare il già citato stato d'ansia) e le 11 modalità di gioco permettono una buona longevità. Arcade Perfect? Quasi, quindi non c'è proprio motivo per non provarlo nella versione per Atari SuperSystem!

di Starfox Mulder

Intervista a Bonaventura Di Bello e Marco Vallarino

Incontriamo i due giornalisti e divulgatori, scrittori e autori di avventure testuali e della nuovissima fiction interattiva intitolata "Déjà Vu": la versione C64 e' disponibile gratuitamente sul sito di RetroMagazine!

di David La Monaca (Cercamon)

INTRODUZIONE

["Fai qualcosa Johnny!" esclama una voce alle tue spalle. "Stiamo precipitando!".

Sei tu Johnny, il pilota di questo aereo? Sembra di sì. Sei seduto proprio davanti ai comandi del velivolo. Eppure ti sembra di non essere qua. Oppure di esserci stato in un altro tempo. Che cosa ti sta succedendo? E' una foresta del Sud America quella che stai sorvolando? Oppure sei nel Borneo? O in Africa? E ti ricordi ancora come si fa a pilotare un simile apparecchio? Tra poco lo scoprirai…]

Fra tutte le opere di intrattenimento che i computer e i dispositivi digitali hanno offerto nel tempo, le avventure testuali (conosciute anche come IF. interactive fiction) hanno sempre rappresentato un genere particolare, molto in voga nella seconda metà degli anni '70, quando i primi monitor dei terminali dei mainframe non erano ancora in grado di mostrare le odierne meraviglie delle animazioni e della grafica ad alta risoluzione. Ma anche qualche anno più tardi, durante tutti gli anni '80, quando i processori video embedded dei sistemi offrivano colori e buone risoluzioni e le schede grafiche avanzate non erano più un'eccezione, le avventure basate solo su descrizioni di testo e comandi più o meno complessi, per interagire con la storia ed i personaggi, continuarono a riscuotere un notevole consenso. Quando sentono parlare di avventure testuali, molti con la memoria risalgono a Zork (1977-1979), a Colossal Cave Adventure (Crowther e Woods, 1976) o al mitico Scott Adams (autore di molti adventure a partire dal 1978), ma già nel 1974 il primo "racconto interattivo" fece capolino col nome di "Wander", scritto da Peter Langston e persino corredato dal primo software in grado di generare altre avventure. In seguito, visto il buon riscontro da parte del pubblico e la disponibilità di strumenti disegnati ad hoc per lo sviluppo di storie interattive, gli autori e le software house interessate al genere si moltiplicarono in tutto il mondo (Italia compresa) dando vita ad un elevato numero di titoli la cui diffusione era garantita da un ormai maturo mercato di home computer sempre più presenti nelle case degli utenti. L'autore di un adventure poteva inserire tutti gli elementi della trama, i dettagli, gli oggetti, le descrizioni dei personaggi e delle ambientazioni e connetterli fra loro per generare un percorso interattivo all'interno del quale il giocatore doveva inoltrarsi per portare avanti la propria sessione di gioco e progredire nella storia attraverso missioni e obbiettivi a breve e lungo termine. Le avventure testuali sono sempre state un genere affascinante e coinvolgente, soprattutto per il fatto che lasciano molto spazio all'immaginazione, come un buon libro, e stimolano la capacità del giocatore nel risolvere piccoli e grandi enigmi, affrontare ostacoli e difficoltà e, in ultima analisi, dipanare una corposa matassa di mistero.

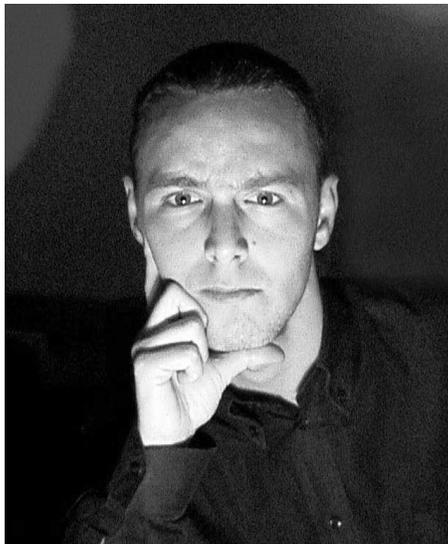
Ne parliamo diffusamente con due fra i maggiori esponenti dell'interactive fiction in Italia, Bonaventura Di Bello e Marco Vallarino, in occasione della recente uscita di "Déjà Vu", un nuovo titolo scritto a quattro mani per il contest "Marmellata d'Avventura" indetto dal sito Old Games Italia.

*I due autori hanno generosamente reso disponibile a tutti i lettori di RetroMagazine la versione di **Déjà Vu** che ha partecipato al contest nell'adattamento per Commodore 64/128 (trovate il link per il download nella sezione Riferimenti).*

Ma andiamo per ordine e cominciamo col conoscerli meglio.

Bonaventura di Bello

L'incontro di **Bonaventura Di Bello (BDB)** – nato a Centola (Salerno) nel 1963 - con le avventure testuali risale al 1986, quando, poco più che ventenne, diventò autore di IF (Interactive Fiction) coronando così in maniera del tutto inaspettata il suo sogno di scrivere narrativa per un vasto pubblico. In poco più di tre anni creò e pubblicò per le collane Explorer, Viking ed Epic 3000 oltre settanta giochi in italiano per le maggiori piattaforme home dell'epoca (C64, ZX Spectrum ed MSX). In seguito, la scrittura diventò professione nel giornalismo, prima per il settore videogiochi con la direzione delle storiche riviste Zzap! Italia e TGM – The Games Machine, poi con diverse altre pubblicazioni informatiche dove non mancava mai il tema del retrocomputing e del retrogaming, fosse anche solo per una rubrica. Oggi, dopo una pausa di qualche anno dedicata alla pubblicazione di manuali tecnici dedicati allo sviluppo per il web, è tornato con piacere al suo 'primo amore', lo sviluppo di IF, e anche stavolta su varie piattaforme, in particolare quelle legate ai dispositivi mobili.

Marco Vallarino (MV) – nato a Imperia nel 1977 - lavora oggi come giornalista presso il quotidiano "Il Secolo XIX" di Genova. Ha pubblicato racconti e romanzi per Mondadori, De Agostini, Multiplayer, Alacran, Addictions, Stampa Alternativa, Fanucci e molti altri editori, ma la sua passione è sempre stata "l'altra" narrativa: quella interattiva delle avventure testuali. "Enigma", "Darkiss" e "Ayon" sono solo alcuni dei suoi giochi più noti, tuttora liberamente scaricabili dal suo sito web www.marcovallarino.it.

Marco Vallarino

L'INTERVISTA DOPPIA

Siete entrambi divulgatori, esperti di comunicazione e da sempre avvezzi ad un utilizzo spinto dei computer per fini ludici e professionali. Che cosa vi ha inizialmente spinto sulla strada dell'informatica e qual è stata la vostra prima esperienza con un computer?

BDB - Quando la diffusione degli home computer ha permesso praticamente a chiunque di acquistare una macchina, la mia curiosità da lettore di fantascienza ha avuto il sopravvento sulle responsabilità familiari (ero poco più che ventenne e già sposato da qualche anno) ma si è configurata anche come una prospettiva professionale. Così mi sono iscritto ad un corso di programmazione su sistemi CP/M cui ha seguito l'acquisto di uno ZX Spectrum. La prima esperienza è stata quindi su macchine CP/M, ma quella determinante è stata di certo quella sulla macchina Sinclair.

MV - Il mio primo computer è stato un Commodore 64, che ho ricevuto come regalo di Natale nel 1986. Dopo aver passato le prime settimane a giocare con i tanti videogiochi presi in edicola e caricati dalle celebri cassettine dell'epoca, un giorno ho avuto un problema al registratore – che mi impediva di caricare i suddetti giochi – e questo mi ha costretto a dare un'occhiata distratta al manuale contenuto nella confezione originale; ho così scoperto il piacere della programmazione (in Basic), che ho portato avanti negli anni in varie occasioni, fino a farla diventare un'attività collaterale a quella giornalistica.

Quando avete deciso che avreste voluto creare qualcosa con un computer invece di limitarvi ad essere un utente di giochi e applicazioni? Avete mai pensato seriamente di cominciare a sviluppare software?

BDB - In realtà l'amore per la programmazione, come combinazione di creatività e logica e di conseguenza come utilizzo di entrambi gli emisferi cerebrali, mi ha affascinato non appena ho iniziato a sbirciare le prime istruzioni del manuale del mio ZX ed i listati, all'epoca sull'enciclopedia Basic di Curcio Editore. Dopo i primi rudimenti appresi nel corso CP/M, il Basic dello Spectrum fu il vero trampolino di lancio per cominciare a creare delle piccole applicazioni anche solo per il gusto di vederle funzionare, ma fu poi l'incontro con un'avventura testuale che fece da innesco per una bruciante passione mai veramente sopita. Il resto, come si dice, è storia.

MV - Negli anni '80, da bambino e ovviamente senza Internet, avevo difficoltà a procurarmi sempre nuovi giochi da provare. Così, nelle pause tra l'uscita in edicola di una cassetta e l'altra, mi mettevo a programmare i giochi "che avrei voluto giocare io", con trame e ambientazioni di solito ispirate ai film, ai fumetti e agli altri videogiochi che mi avevano colpito di più. Molti anni più tardi, l'avvento di Internet mi ha permesso di condividere con un pubblico sufficientemente vasto (almeno per le mie aspettative) i giochi e i racconti che da sempre tenevo nel cassetto o che avevo mostrato a pochi amici, di solito scarsamente interessati. Il buon riscontro ottenuto fin da subito – per esempio gli oltre 50.000 download ottenuti su Tuttogratis da "Il giardino incantato" – mi ha convinto a scrivere nuove opere, sia interattive che non, che negli anni mi hanno permesso di trasformare la passione per la scrittura in un lavoro (sono dal 2005 un giornalista a tempo pieno) e quella per i videogiochi in un'attività che ha avuto declinazioni didattiche e promozionali di un certo rilievo – cito per tutti il videogioco "Visita al Marconi" finalista nel 2016 agli Italian Gamification Awards.

Due autori come voi hanno sicuramente sviluppato la propria creatività sulla base di esperienze dirette. Qual è stato a grandi linee il vostro percorso di studi ed aveva a che fare con l'uso di computer?

BDB - Nella mia infanzia e adolescenza sono stato soprattutto un avido lettore, soprattutto di fantascienza ed è stato l'amore per la lettura e lo studio/ricerca in generale e un'infinita curiosità a permettermi di continuare la mia 'formazione' da semplice autodidatta, negli anni successivi, parallelamente all'ingresso nel mondo professionale che avvenne proprio negli anni in cui molti miei coetanei erano ancora 'ospiti' dei genitori e studenti universitari, mentre io cominciavo ad avere già la responsabilità di una famiglia essendomi sposato a vent'anni. Ho intrapreso per ben due volte il percorso universitario, abbandonandolo dopo un anno sia per delusione rispetto al programma didattico sia per mancanza di tempo e fondi, ma non ne sono affatto pentito visti i risultati.

MV - Dopo aver trascorso i primi mesi di fortunato e felice possessore di Commodore 64 a giocare a classici giochi di piattaforme, sparatutto, sport e azione, ho ricevuto in regalo una cassetta con alcuni strani giochi in cui al posto del joystick bisognava utilizzare la tastiera: le avventure testuali, che all'epoca si chiamavano semplicemente adventure. L'inizio fu ostico, ma in poche ore mi feci prendere dalle storie, i testi e le illustrazioni di quei giochi così realistici per l'epoca, che mi permettevano di vivere da protagonista vicende d'azione, d'avventura, di mistero che avevo visto solo in pochi film. Non m'importava se c'era poca grafica, se non c'era musica e se si poteva rimanere bloccati per giorni – all'epoca anche per mesi! – in qualche punto oscuro della storia. Per me il bello era poter giocare DENTRO quelle storie come mai ero riuscito a fare prima. In breve

dedicai le mie primitive (ma sufficienti per lo scopo) conoscenze di programmazione in Basic per scrivere avventure mie da mostrare con orgoglio ad amici e parenti. Iniziò così l'Avventura con la A maiuscola che continua tutt'oggi e che ha condizionato gran parte della mia vita professionale e in un certo qual modo anche quella privata. Naturalmente gli adventure che mi furono regalati erano quelli scritti da Bonaventura Di Bello per la rivista Explorer!

Qual è stato il vostro sistema preferito fra quelli di un tempo (dagli home computer in avanti) e perché? (Potete anche aggiungere la descrizione dell'intero sistema che possedevate).

BDB - Naturalmente la prima macchina rimane nel cuore di tutti noi perché, appunto, è quella su cui abbiamo compiuto i primi passi. Nel mio caso fu, come ho già detto, lo ZX Spectrum di Sinclair, che diventò presto uno strumento di lavoro dopo essere stato il pretesto per scoprire le AT grazie ad una espansione di memoria (dai 16KB di base a 48KB).

MV - Nel 1989, il regalo per la promozione a giugno dalla prima alla seconda media è stato il Commodore Amiga 500, computer a 16 bit su cui sognavo già da un paio d'anni di mettere le mani per via della grafica strabiliante e della maggiore velocità e potenza di calcolo rispetto al Commodore 64. Lo tenni fino al 1993, quando – come tanti altri teenager dell'epoca – passai al mondo dei PC. Essendo ormai diventato ragazzo, con l'Amiga 500 potei vivere la passione per l'informatica in maniera più consapevole e matura e quindi scrivere giochi più complessi e adatti a una condivisione con il pubblico, anche se – come detto – fu solo l'avvento di Internet che mi permise di far provare le mie avventure a persone che non fossero amici o parenti. Quello dell'Amiga lo ricordo come un periodo particolarmente felice e florido della mia vita. Riuscivo a farci tutto quello che volevo senza particolari sforzi o stress. Il passaggio al PC portò anche lo shock che il listato in AmigaBasic che usavo per scrivere le mie avventure – attività pressoché settimanale – non funzionava in QBasic; per adattarlo mi ci volle un po' e questo contribuì a creare un effetto nostalgia che negli anni si è solo attenuato.

Qual è stata l'ispirazione principale che vi ha portato a sviluppare interactive fiction e in particolare le vostre prime avventure testuali?

BDB - La scintilla che scatenò la passione bruciante per l'IF fu una "fregatura": il commesso del negozio che doveva darmi un gioco per 48K in regalo per il mio acquisto di espansione di memoria dello Spectrum pensò bene di rifilarmi un gioco che non riusciva a vendere, ovvero un'avventura testuale per Spectrum 16K. Dopo aver trascorso le settimane successive a giocarla e risolverla, decisi che sarei diventato un autore di AT.

Riviste Epic3000 e Viking

MV - Come detto, l'impatto con gli adventure scritti da BDB per le collane Explorer e Viking fu travolgente: in breve, tra un'uscita e l'altra delle due riviste, quando rimanevo a secco di storie interattive, iniziai a programmarne di mie, senza troppe pretese, ma con un entusiasmo facilmente immaginabile. La prima in assoluto fu una specie di poliziesco, "Fuga dal carcere"; la seconda un fantasy, "Mostri e magie", in cui c'era una zucca di Halloween che moltissimi anni dopo ebbi modo di riciclare in "Darkiss 2". Poi iniziai a scrivere anch'io, come BDB, avventure dedicate a personaggi particolari: il primo fu il giornalista del mistero (che potrebbe essere collega di Roy Norton) Jack Newton, recentemente riesumato come assistente di George Anderson in "Sogno di Sangue".

Qual è stata invece la prima avventura testuale che avete mai giocato e qual era la vostra tecnica di gioco (prendevate appunti, disegnavate mappe su carta, ecc.)?

BDB - L'avventura per ZX 16KB di cui parlavo era "Adventure A: Planet of Death" di Artic Computing e istintivamente cominciai, dopo le prime frustrazioni (non tanto per la lingua inglese, che già conoscevo bene, quanto per lo schema di gioco) a tracciare una mappa su carta dove annotavo anche eventuali indizi. Questo mi permise infatti di superare le fasi più contorte e complesse del gioco.

MV - "La valle incantata", scritta da BDB per il numero 4 di Explorer, la rivista con cassetta che mi fu regalata per il mio onomastico e che mi permise appunto di entrare nel magico mondo dell'avventura. Ricordo che la giocai e finii senza disegnare una mappa o prendere appunti: già all'epoca, come oggi e forse anche di più, avevo una memoria eccezionale (mia madre dice che è merito di tutto il pesce che mi ha fatto mangiare da bambino, chissà!) e riuscivo a ricordare tutti i particolari necessari per andare avanti nell'azione. Quando rimasi bloccato, come tutti, chiamai il numero dell'help line messo sul giornale e quello fu il primo incontro (almeno telefonico) con quello che sarebbe diventato il mio nume tutelare e oggi coautore di "Déjà Vu": Bonaventura! Tornando a "La valle incantata", ancora oggi lo considero un fantasy paradigmatico per la disposizione degli ambienti di gioco, dei puzzle da risolvere e delle scene d'azione. "Il giardino incantato" gli è debitore per molte cose, immagino si capisca anche dal titolo.

Quale sistema di produzione di avventure testuali avete usato soprattutto all'inizio? Avete cominciato con il Basic o con un altro linguaggio di programmazione per le vostre prime sperimentazioni?

BDB - Dopo aver testato vari sistemi ("Dungeon Builder", "Graphic Adventure Creator" e qualche altro di cui ormai ho dimenticato il nome) mi innamorai di "The Quill", un tool sviluppato dall'azienda gallese Gilsoft (l'autore era Graeme Yeandle, se ricordo bene). A questo si affiancò quasi subito "The Illustrator", che permetteva di includere illustrazioni con una rudimentale, ma efficace, grafica vettoriale. Per le avventure su MSX, sviluppate in seguito, adattai il famoso "modulo Basic" di Enrico Colombini in modo che ricalcasse la struttura dati di The Quill, ovviamente senza grafica a causa delle limitazioni di memoria del sistema.

MV - Ho cominciato a programmare con il Basic del Commodore 64, come credo tanti altri bambini dell'epoca e proprio il desiderio di scrivere avventure testuali simili a quelle che compravo in edicola mi ha permesso di misurare e migliorare le mie capacità di programmatore (in erba). Nel 1990, dopo essere passato all'Amiga 500, ho trovato sulla rivista Amiga Byte un corso per programmare avventure testuali scritto da Maurizio Giunti, che mi ha consentito di realizzare altri giochi, tra cui un kolossal – per i miei standard – di oltre 80 stanze in cui il cattivo della storia era addirittura Dracula! Nel 1995, da utente PC, mi procurai uno dei manuali di Enrico Colombini per ricominciare a scrivere avventure, sempre in Basic: in quel periodo, ero al liceo, scrissi sia "Sfida all'ignoto" che "Il giardino incantato", due giochi non certo curati e complessi come "Ayon" o "Darkiss", ma che ancora oggi hanno il loro pubblico.

Qual è l'avventura più sorprendente e coinvolgente, non necessariamente lontana nel tempo, cui avete mai giocato? Perché la ritenete la migliore mai pubblicata?

BDB - Sicuramente "Lurking Horror" di Infocom, anche se in realtà tutte le avventure giocate all'epoca, dalle più piccole a quelle più complesse e articolate, sono state fonte di sfida e divertimento ben più di qualsiasi altro videogame.

Marco Vallarino durante una conferenza

MV - Sicuramente "Acheton", un'avventura nata in Inghilterra alla fine degli anni '70 nel mondo dei mainframe universitari – che peraltro fece da sfondo a tante altre eccellenti creazioni videoludiche – e poi verso la metà degli anni '80 rilasciata al pubblico dalla Topologika in una versione commerciale. Opera di Jon Thackray, David Seal e Jonathan Partington, è una avventura che vanta oltre 400 locazioni da mappare accuratamente, 55 tesori da trovare (e da sistemare in una cassaforte grande come una stanza!), innumerevoli scenari da esplorare in lungo e in largo: foreste, case, giardini, serre, pozzi, cimiteri, caverne, cunicoli, sotterranei, labirinti, miniere, ghiacciai, deserti, piramidi, canyon e isole. C'è perfino il nido di un uccello Roc! Ci ho messo sette anni a finirla, ma arrivare nella "Gladiators' Arena" e vincere tutti i combattimenti è stata una delle più grandi soddisfazioni della mia 'carriera' di avventuriero. Al di là della grande varietà dell'ambientazione e dell'azione di gioco, quello che mi piace di "Acheton" è che, sia pure con un grande sforzo creativo e tecnico, è un'opera che offre al giocatore un intero mondo da esplorare e quindi un'esperienza immersiva totale.

Quando avete programmato le vostre avventure più "mature", avete utilizzato uno specifico editor? Se sì, potete dirci quale e perché l'avete scelto fra gli altri? Qual è stato il metodo che avete adottato per passare dal concept di base allo sviluppo vero e proprio delle avventure?

BDB - Come dicevo, l'adozione di "The Quill" si rivelò una scelta decisiva, in quanto essendo un sistema RAD (Rapid Application Development) permetteva di concentrarsi su un meta-linguaggio in cui venivano definite le azioni e gli eventi e su tabelle di dati (oggetti, locazioni e loro descrizioni e riferimenti). Ciò mi consentì di non perdere tempo sulla 'codifica' pura di un linguaggio di programmazione, ma piuttosto di concentrarmi sulla parte creativa (il testo) e quella logica (gli enigmi), creando fino a sei avventure in un mese e 'portandole' contemporaneamente su tre diversi sistemi, in quanto potevo focalizzarmi sulla trama, sui personaggi e sui luoghi e oggetti con cui il giocatore avrebbe interagito, elaborati direttamente attraverso mappe 'analogiche' corredate da appunti. Qualcuno, probabilmente per suoi sentimenti personali, tempo fa scrisse in un forum che "producevo avventure con lo stampino", ma posso dire di essere più che soddisfatto della mia produzione di allora, soprattutto per i riscontri che ho avuto fra il pubblico dell'epoca e persino da qualche giocatore "maturo" odierno (qualcuno mi contatta ancora oggi per farsi aiutare nella soluzione di quei giochi!). Oggi utilizzo strumenti digitali in quanto lavoro soprattutto su iper-narrativa (le cosiddette 'storie a bivi'), ma lo sviluppo di ogni titolo è preceduto da ore e ore di full immersion con trascrizione di appunti su carta.

MV - Nel 2002 sono passato dal Basic a Inform 6, il linguaggio orientato alla creazione di videogiochi testuali creato da Graham Nelson, perché mi sembrava interessante la possibilità di creare un gioco che, tramite appositi interpreti, fosse utilizzabile anche al di fuori di Windows (che all'epoca costituiva il 99% del mercato). Qualche anno dopo, la diffusione di smartphone e tablet e l'affermazione di sistemi alternativi come GNU/Linux e Mac OSX, ha confermato che il formato multi piattaforma era il migliore su cui puntare. Di solito, quando mi viene un'idea per un gioco nuovo, inizio a stilare un elenco dei luoghi e delle cose che "dovrebbero esserci", sottolineando quelli che mi sembrano più adatti per muovere il gioco, dare personalità alla storia e rendere l'azione più complessa. Poi penso agli oggetti e ai personaggi necessari per dare vita all'avventura. Dopo di che disegno la mappa della locazioni con quello che ci dovrebbe essere dentro o dovrebbe accadere. La cosa più difficile di solito è tarare la difficoltà dei puzzle di modo che all'inizio siano più facili e alla fine più difficili: non sempre la storia lo permette perciò bisogna scegliere se prediligere l'aspetto ludico o quello narrativo.

Quali sono state le difficoltà principali che avete incontrato nello sviluppo e nell'eventuale porting delle vostre avventure verso altri sistemi a 8/16 bit rispetto a quello iniziale?

BDB - L'uso di uno strumento multi-piattaforma come "The Quill" (che in ogni caso richiedeva la riscrittura del meta-codice non essendoci un convertitore automatico) mi permise di dover affrontare il problema del vero e proprio porting solo quando si palesò la necessità di sviluppare su MSX, dove come dicevo fui costretto a usare il Basic. Oggi non mi occupo più della programmazione, in quanto si lavora in team e lo sviluppo del codice è affidato a qualcun altro, ma utilizzo comunque un sistema di meta-dati oppure codice "universale" come XML o JSON per

fornire al programmatore tutto quanto occorre per produrre un gioco in grado di funzionare su tutte le piattaforme previste dal progetto.

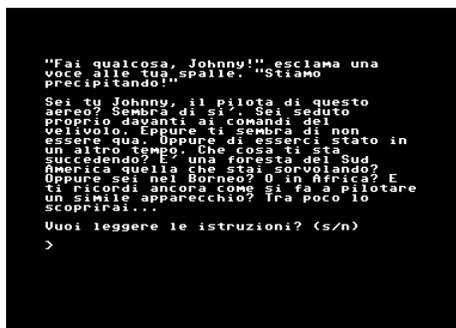
MV - Una certa parte della mia attività di game designer è sempre stata dedicata alla portabilità e all'accessibilità dei miei giochi. Il riscontro di pubblico (e qualche volta di critica) è per me molto importante: questo è uno dei motivi per cui – come detto – nel 2002 sono passato dal Basic a Inform 6, che permetteva di proporre i propri giochi in un formato più "gradito" al pubblico di allora (e anche di oggi, mi pare). Non ricordo particolari difficoltà nel processo di conversione, anzi, si è trattato di un lavoro che mi ha permesso di rendere certi giochi migliori o almeno più semplici. L'unica cosa che mi manca del QBasic è la routine che avevo scritto per "Enigma" e che permetteva di muoversi anche con i tasti cursore, che invece Inform 6 riserva all'editing della riga di comando.

Possedete ancora e utilizzate di tanto in tanto sistemi retro o addirittura lo stesso hardware che avete utilizzato tanti anni fa per giocare o per creare le vostre avventure?

BDB - Per problemi di spazio e mobilità (sono stato per anni quel che si dice un "nomade digitale") non ho avuto la possibilità di continuare a usare le vecchie macchine e mi sono pertanto limitato agli emulatori. Ovviamente la tentazione di riesumare retro-hardware di tanto in tanto mi coglie, ma riesco a sopirla partecipando a qualche evento di retro-computing dove le macchine di allora sono presenti e funzionanti.

MV - Ho ancora sia il Commodore 64 che l'Amiga 500, ma nel 90% dei casi gioco o programmo con gli appositi emulatori anziché con i computer dell'epoca. Grazie all'ottimo sito di Jacob Gunness www.solutionarchive.com (noto anche come CASA o Classic Adventure Solution Archive) negli anni ho scoperto e giocato molte vecchie avventure, alcune delle quali mi sono servite da spunto per le mie. Come diceva Picasso: un grande artista non copia, ruba!

Molti considerano Zork o Colossal Cave Adventure (1976) i precursori del genere adventure ma di recente proprio BDB ha scoperto che il vero antesignano sarebbe un gioco (corredato anche da un editor di AT) chiamato "Wander" realizzato su mainframe PDP-10 nel 1974. Dopo il successo delle avventure testuali, che cosa avete pensato delle avventure che includevano anche delle immagini grafiche statiche e poi di quelle animate "punta e clicca", da Maniac Mansion e Zak McKracken in avanti?

BDB - Per quanto mi riguarda, pur avendole in parte apprezzate per le trame e gli elementi di gioco, hanno sempre rappresentato nella mia percezione la stessa differenza che c'è fra un libro e un fumetto, ovvero hanno "rubato" al giocatore l'emozione di immaginare luoghi, personaggi e situazioni in modo del tutto personale e intimo, creando un immaginario individuale che solo il testo riesce a evocare. Capisco, tuttavia, che il mercato ha le sue esigenze, soprattutto quella di raggiungere un pubblico sempre più vasto, ma sono certo che c'è ancora spazio per una IF dove l'equilibrio fra testo e immagini sia tale da non precludere l'esperienza del lettore/giocatore e non penalizzare la sua capacità immaginativa.

Deja Vu - Screenshot dal C64

MV - Sono stato un fan della prima ora sia di "Zak McKracken" sia di "Maniac Mansion" e "Indiana Jones and the Last Crusade" (ho ancora la mia copia del National Inquisitor e altri pregevoli gadget dell'epoca). Tuttavia ho sempre considerato un'avventura grafica come un altro genere di gioco anziché – come si diceva all'epoca – come un'evoluzione di quella testuale. Mi dispiace che le avventure testuali siano andate fuori mercato quando c'è stato il boom di quelle grafiche perché secondo me potevano continuare a esistere (e progredire) in due contesti differenti, oltre che promuoversi a vicenda. D'altra parte le stesse avventure grafiche basate su SCUMM [n.d.A. – SCUMM è il linguaggio ed ambiente di programmazione interattivo ideato dalla Lucasfilm per tutte le sue avventure grafiche, acronimo di Script Creation Utility for Maniac Mansion) non è che siano durate molto di più, salvo poi diventare protagoniste di varie "operazioni nostalgia" che hanno confermato come certi fenomeni non fossero legati solo a mode del momento. Oggi, le avventure commerciali non esistono quasi più e mi pare che se ne senta la mancanza.

Cosa pensate del mondo del retrocomputing e del relativo successo che questo fenomeno sta avendo in questo periodo (vedi remake di giochi e vendita di nuovo hardware tipo console "mini", board emulators, nuovi accessori e periferiche moderne per i vecchi sistemi)?

BDB - Probabilmente si tratta di un fenomeno legato al fatto che gli ex-ragazzi di allora oggi sono degli adulti con maggiori capacità di acquisto e una forte componente nostalgica e in ogni caso penso che si tratti di un fenomeno legato a una passione tecnologica non diversa da quella per l'hi-tech, se non per le sue connotazioni storiche. Mi chiedo, più che altro, se un fenomeno del genere avrà mai una traslazione nel futuro, ovvero se ci sarà mai, nei decenni che verranno, qualcuno che per esempio farà del retrocomputing con le console che si utilizzano oggi. Non riesco a immaginarlo, tuttavia, se devo essere sincero.

MV - Come dicevo prima, certi fenomeni hanno avuto un impatto talmente grande sulla gente che non potevano esaurirsi nel loro (troppo) breve arco commerciale. Appena entrato in Internet, nel 1998, gli emulatori dei miei vecchi computer e console, e le rom dei giochi, sono stati le prime cose che ho cercato. Del resto, ci sono libri e fumetti che leggiamo e rileggiamo, e film che riguardiamo, anche a distanza di anni dalle prime volte. O canzoni che riascoltiamo, luoghi che torniamo a visitare, eccetera. Anche perché ci sono giochi il cui valore non è limitato dalla potenza della macchina sulla quale erano stati realizzati: le avventure testuali sono il caso più emblematico, ma per me "Kick Off 2" dell'Amiga 500 è ancora oggi il migliore gioco di calcio della storia.

Frequentate qualche gruppo FB o forum specifico sulle IF o sui retrocomputer? (Specificate pure quali)

BDB - Di tanto intanto lurko in qualche forum come quello di ODG (Old Games Italia), dove sono stato invitato tempo fa dopo il mio ritorno all'IF con l'iper-narrativa, ma il mio rapporto con FB è cessato da tempo, in quanto ho percepito dopo qualche anno i lati negativi della piattaforma e mi sono così limitato a mantenere un account solo per lo scambio di messaggi personali. Preferisco, invece, consultare siti italiani dedicati all'IF o portali come Ready64.org oppure RetroEdicola Videoludica, come anche quelli stranieri più noti, spesso raggiunti attraverso ricerche mirate.

MV - Su Usenet seguo il newsgroup it.comp.giochi.avventure.testuali dalla sua fondazione, cioè dal 1999. Su Facebook, oltre al gruppo di supporto della mia avventura "Darkiss" (in cui peraltro si parla di molti altri giochi), seguo e in parte gestisco la pagina delle "Avventure testuali" insieme agli amici di Oldgamesitalia.net.

Qual è stato il vostro coinvolgimento con le società di software che nel tempo hanno pubblicato le vostre opere? (Avete scritto da soli tutte le storie, le sceneggiature e le dinamiche di gioco oppure c'erano altri autori che hanno sviluppato le trame insieme a voi? Vi hanno posto limiti in fase di ideazione o di produzione?)

BDB - Per quanto mi riguarda ero totalmente responsabile dello sviluppo, tanto dei contenuti quanto del codice, fino alla produzione del "nastro" finale. Ero persino responsabile dei contenuti che andavano a formare la componente cartacea abbinata alla cassetta. Non ho mai avuto paletti particolari se non alcune indicazioni, preziosissime, che l'editore mi fornì in modo da rendere più appetibili le storie (per es. la scelta di un personaggio che tornasse in avventure successive). Le illustrazioni delle copertine cartacee, invece, erano affidate ad un grafico, e sono sicuro che abbiano avuto un ruolo importante nell'appetibilità di quei titoli, come lo è la copertina di un libro dopotutto.

MV - A parte questa già leggendaria collaborazione con BDB per "Déjà Vu" devo ammettere che ho quasi sempre fatto tutto da solo. Negli anni mi sono arrivate diverse proposte di collaborazione per progetti anche interessanti, ma mi trovo più a mio agio a lavorare in proprio, almeno per quanto riguarda il design. Tuttavia sto da tempo portando avanti un progetto nel quale, grazie al fatto che i ruoli sono ben delineati e distinti, sto rivalutando la forza del lavoro di squadra. Poi è stato interessante, qualche anno fa, preparare delle versioni speciali dei miei giochi di punta, "Darkiss", "Enigma" e "Ayon", per le riviste di informatica – Linux Pro, Win Magazine, Computer Bild, Idea Web e The Games Machine – che erano interessate a pubblicarli nei loro CD e DVD allegati. Tra le richieste più frequenti c'era quella di allegare al programma una documentazione sufficientemente chiara e completa perché gli utenti potessero giocare senza dover inondare la redazione di richieste di aiuto e informazioni – essendo le avventure testuali un genere abbastanza codificato e talvolta ostico come approccio.

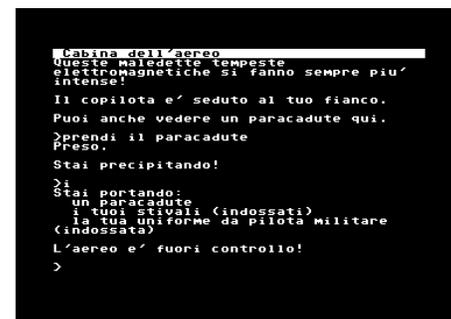
Se ci fossero una o più cose che poteste cambiare o completare in una delle vostre avventure o nel processo creativo, tornando indietro nel tempo, quali sarebbero?

BDB - Indubbiamente, se avessi avuto più tempo, [cambierei] la complessità delle storie (come numero di elementi in esse contenuto) e la lunghezza dei testi. E magari l'inserimento di un maggior numero di "aiuti", anche se all'epoca compensavo con la famosa help-line telefonica.

MV - In realtà è una cosa che ho già fatto. Giochi come "Il giardino incantato" e "Sfida all'ignoto" sono stati riscritti e riprogrammati più volte nel corso degli anni, per adattarli al periodo oltre che per correggere piccoli e grandi errori. Anche "Darkiss" e "Ayon" sono stati oggetto di restyling. Per me un gioco, così come un racconto o un articolo, non è mai definitivo al 100%: se la perfezione – come si dice – non è di questo mondo, dobbiamo sempre essere pronti a fare di meglio.

Quali consigli dareste a coloro che oggi si avvicinano a questa categoria di giochi o d'intrattenimento digitale come potenziali autori?

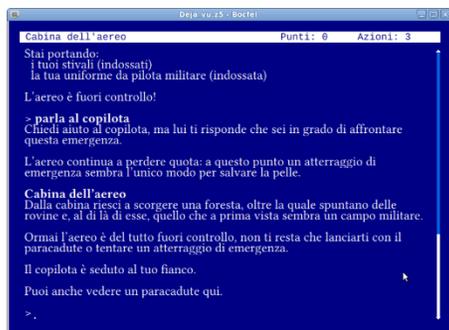
BDB - Sicuramente consiglierei di non trascurare mai l'elemento emozionale, ossia di chiedersi sempre cosa vogliamo evocare nel giocatore/lettore e qual è il modo migliore per farlo. Oltre ciò, è bene non smettere mai di alimentare la propria fantasia con letture e film, e di tenere presente le tendenze di fruizione del momento, che ovviamente cambiano a seconda delle tecnologie e dei periodi storici.

```
 Cabina dell'aereo
Queste maledette tempeste
elettromagnetiche si fanno sempre piu'
intense!

Il copilota e' seduto al tuo fianco.

Puoi anche vedere un paracadute qui.

>prendi il paracadute
Preso.

Stai precipitando!

>i
Stai portando:
  un paracadute
  i tuoi stivali (indossati)
  la tua uniforme da pilota militare
(indossata)

L'aereo e' fuori controllo!

>
```

Deja Vu - Screenshot dal C64

MV - Il consiglio lo dò da giocatore prima che da autore: andateci piano. Creare giochi super-difficili e lunghi non vi farà avere più pubblico o complimenti, farà solo infuriare la gente che rimane bloccata o che non arriva mai alla fine della storia. Nell'era digitale, l'immediatezza è la strada da seguire: bisogna essere chiari e concisi per conquistare l'attenzione dell'utente e mantenerla fino in fondo. E poi ovviamente bisognerebbe cercare di conoscere un po' il "mercato", per capire che cosa funziona (e magari potrebbe funzionare meglio) e che cosa manca. E poi occorre ricordarsi che il lavoro non finisce con la pubblicazione del gioco, ma continua con la fase di promozione che richiede altrettanto impegno se non di più.

Avevate mai immaginato che le vostre avventure avrebbero raccolto un così largo apprezzamento dagli utenti di sistemi così differenti?

BDB - A dire il vero inizialmente avevo i miei dubbi, in quanto ero consapevole che il trend maggiore sarebbe sempre stato quello dei videogiochi classici (shoot'em up, soprattutto), e sono rimasto piacevolmente sorpreso non solo di scoprire un interesse tanto vasto nell'IF in quel periodo, ma anche la "sopravvivenza" di tale interesse in una considerevole parte del pubblico nei decenni successivi.

MV - In verità oggi mi sembra più normale che un'opera abbia un pubblico piuttosto che non ce l'abbia. Ormai ci sono così tanti spazi di promozione che non vedo come un autore non possa trovare almeno qualcuno interessato al suo lavoro. Se però si vuole parlare di grandi numeri, picchi di ascolto (e di download), stagioni d'oro e d'argento, probabilmente certi exploit sono dovuti a un mix di perseveranza e ingegno (e fortuna) che ogni creativo dovrebbe avere, oggi, se è abbastanza motivato e supportato, da amici, colleghi e dalla stessa comunità che ha contribuito a creare con la sua opera. Quando ho cominciato a scrivere avventure, quello che non sapevo è che i videogiochi testuali hanno un grande riscontro tra gli utenti non vedenti, che possono utilizzarli grazie alla sintesi vocale. Questo, come detto, mi ha portato a lavorare con attenzione sul fronte dell'accessibilità, per permettere a tutti di giocare. Negli anni, come con altri utenti, sono nate delle belle amicizie e il fatto che il mio lavoro possa anche avere uno scopo sociale – cioè permettere di giocare a chi è impossibilitato a cimentarsi con altri tipi di videogiochi – è gratificante.

Deja Vu - Screenshot della versione PC

I formati digitali come ePub e Kindle rappresentano secondo voi una strada percorribile per i cosiddetti "fiction games"? E se sì, avete già pensato di fare esperimenti su questa piattaforma?

BDB - Sì, ho infatti pubblicato un manuale dedicato ("Iper-Narrativa") e un ebook demo gratuito ("Il Segreto dei Wilkinson") disponibile su tutte le piattaforme e librerie on-line, quest'ultimo con il solo obiettivo di mostrare il funzionamento di una storia "a bivi", piuttosto che pretendere di esordire con un'opera di narrativa vera e propria. Il risultato, nel caso degli ebook, è strettamente legato al metodo di progettazione, che ho appunto esplorato e descritto nel mio manuale e discusso anche sul mio blog. Una volta pianificata correttamente ed efficacemente la trama e relativi "bivi", lo sviluppo è abbastanza semplice considerati gli strumenti oggi a disposizione di chi volesse cimentarsi in questo genere di produzione ludico-narrativa.

MV - Sono un felice possessore di Kindle Paperwhite da vari anni, ma credo che smartphone e tablet (oltre che pc) siano i dispositivi migliori per fare da base al mercato dei cosiddetti casual game, nel quale credo possano rientrare anche le interactive fiction. D'altra parte, non so quale futuro possa esserci per gli ebook reader ora che con gli smartphone si può fare tutto ed esiste pure un'app che simula il funzionamento di un ebook reader.

Quali punti di contatto ci sono secondo voi fra il genere ormai così diffuso delle Graphic Novel e le Interactive Fiction? Potrebbe nascere secondo la Vostra esperienza un genere letterario che coniughi fumetti e interattività su tablet o PC?

BDB - Certamente ed è infatti proprio il campo su cui sto lavorando da tempo, ma è importante appunto tener conto delle tendenze e delle abitudini di fruizione del pubblico attuale, sempre meno disposto a dedicarsi intensivamente a un prodotto che metta soprattutto in gioco le capacità intellettuali e deduttive piuttosto che i riflessi e la velocità di esecuzione.

MV - Come detto, anche per non limitarne l'accessibilità, ho sempre lavorato a progetti di testo puro – tranne nel caso di "Visita al Marconi", ma quello era un gioco (didattico) su commissione. La possibilità che in un mio gioco ci siano immagini o suoni di corredo non mi accende più di tanto. Sono sempre stato un avido lettore di fumetti e sono sicuro che si possa realizzare una bella interactive fiction a fumetti, ma in quel caso immagino sarebbe meglio pensare a un'avventura grafica a tema. (E probabilmente ce ne sono già state.)

Parliamo ora di "Déjà Vu", il nuovo titolo che avete deciso di sviluppare insieme in occasione del contest "Marmellata d'Avventura" indetto da Old Games Italia.

Com'è avvenuto il vostro incontro e a chi è venuta l'idea del soggetto?

BDB - In realtà il nostro "incontro" avvenne proprio negli anni Ottanta, quando Marco venne a trovarmi in redazione, ancora bambino, insieme a suo padre. Ci siamo ritrovati in seguito e poi tenuti in contatto, con la promessa di sviluppare prima o poi almeno un'avventura assieme; e il pretesto del contest "Marmellata d'Avventura" è stato, insieme a una mia temporanea disponibilità di tempo in quel periodo, il fattore scatenante che ha dato vita, in un tempo relativamente breve, a questa collaborazione che devo dire mi ha coinvolto moltissimo e mi ha fatto estremamente piacere. Il soggetto nasce soprattutto da una serie di letture e film del passato, oltre che dalle esperienze di gioco delle AT, ed è infatti in gran parte rievocativo e celebrativo (qualcuno ha notato l'incipit e la sua affinità con l'avventura di Enrico Colombini, cui personalmente devo molto per l'aiuto nel porting su MSX delle mie avventure). Io mi sono dedicato a sviluppare soprattutto la storia e i suoi elementi, compresa la mappa con oggetti ed enigmi, dopodiché Marco ha arricchito il tutto con le descrizioni, i messaggi e ovviamente la programmazione in Inform che io non avrei avuto né il tempo né la capacità di affrontare, visti i miei impegni e competenze attuali.

MV - Quando a febbraio ho saputo del contest "Marmellata d'avventura", mi sono iscritto subito alla gara, immaginando che la gente si aspettasse una mia partecipazione. Tuttavia, non avevo particolari idee o progetti in mente – del resto il tema non era ancora stato reso noto. In ogni caso, puntavo a "fare presenza" per non deludere il mio affezionato pubblico, ma niente di più perché prevedevo che marzo sarebbe stato un periodo "pieno" al giornale e comunque temevo che in un mese si sarebbe potuto fare poco. Pochi giorni prima della scadenza del bando, BDB mi ha scritto dicendo che gli sarebbe piaciuto partecipare ma gli serviva che qualcuno gli programmasse (e quindi adattasse) il gioco. Per me, come per tutti, era una notizia EPOCALE: il maestro Di Bello voleva scrivere una nuova avventura, trent'anni dopo quelle di Explorer e Viking. Accettai dandogli carta bianca sul soggetto, con la curiosità di sapere dove sarebbe andato a parare – tuttavia, feci anche a lui l'invito a "andarci piano" perché "non sono più gli anni

80 e la gente non ama morire senza preavviso o rimanere bloccata senza sapere perché".

Come avete impostato il lavoro, il processo creativo e la trama? A proposito, potete anticiparci qualcosa dei personaggi, dell'ambientazione e della storia?

BDB - Il lavoro si è basato inizialmente su uno scambio di idee e proposte attraverso un semplice documento di testo condiviso in Google Docs, dopodiché io ho proseguito lo sviluppo della mappa e degli elementi di gioco usando un tool comodissimo che si chiama Trizbort, con il quale ho potuto poi passare a Marco le informazioni (sotto forma di mappe e meta-dati) di cui aveva bisogno per proseguire nello sviluppo, ovviamente corredate dagli appunti nel documento condiviso di cui sopra e da una fitta serie di conversazioni a voce o via chat che hanno accompagnato lo sviluppo ed il testing. Per quanto riguarda la storia, l'ambientazione e i personaggi diciamo che gli ingredienti sono più "avventurosi" che mai, e il tutto si svolge in un contesto distopico sicuramente molto caro al pubblico attuale, ma che il finale sarà quasi certamente "rivelatorio" e inaspettato per la maggior parte dei giocatori, come è giusto che sia in ogni storia degna di essere giocata o letta.

MV - La sera di venerdì 16 marzo, quando ormai oltre metà del tempo a disposizione per scrivere il gioco era passata, ho ricevuto l'email con cui BDB mi mandava mappa e trama del gioco: c'erano più di quaranta locazioni e decine di azioni da adattare e programmare. Superato lo shock iniziale per la quantità di lavoro che c'era da fare in due settimane (scarse), mi addentrai nella lettura e capii che era un progetto che meritava davvero il massimo impegno, sia pure – come detto – dopo qualche adattamento. Scrivere (quasi) tutti i testi del gioco mi ha permesso di entrare in piena sintonia con lo spirito della storia, che credo rappresenti il giusto connubio tra passato e presente del genere avventuroso. Presentata come "un'avventura ai confini del mondo e della mente", Déjà Vu ha per protagonista il pilota di un aereo militare che deve portare a termine una certa missione. Il volo però non si conclude come previsto e per Johnny inizierà una "nuova" avventura, in cui le sorprese non mancheranno.

Su quale editor e con quali strumenti sono avvenuti avvenuta la stesura e lo sviluppo vero e proprio? Per quali sistemi sarà disponibile la vostra opera, almeno inizialmente?

BDB - Come dicevo abbiamo optato per Inform, sia perché Marco ormai lo conosce bene, sia per la possibilità di produrre un codice abbastanza universale, ma devo dire che Marco mi ha stupito successivamente inviandomi persino una versione per C64! Ci tengo a precisare che "Déjà Vu", almeno nel formato attuale, è semplicemente una storia di base destinata al contest, con i limiti inevitabili che il tempo ridotto e la destinazione d'uso impongono. Abbiamo, tuttavia, pensato di produrne una versione ampliata e molto più ricca dal punto di vista dei contenuti e dello schema di gioco, sotto forma di app per dispositivi mobili, in collaborazione con un programmatore che si è dimostrato interessato al progetto. Chissà che non rappresenti il primo di una serie di giochi di IF da proporre al pubblico nel medio-lungo termine.

MV - "Déjà Vu" è stata scritta in Inform 6, come "Darkiss", "Ayon" e le altre mie avventure disponibili per il download. Grazie al formato multipiattaforma fornito dall'estensione Z-code (z5 in questo caso), può essere giocata su quasi tutti i sistemi operativi esistenti (anche su Commodore 64 e Amiga 500, per dire), utilizzando uno degli appositi interpreti gratuiti. Nel file zip che si può scaricare dal sito www.marcovallarino.it c'è tutto ciò che serve sapere per iniziare a giocare.

Cosa pensate del panorama odierno dei videogiochi in generale e del settore dei giochi interattivi in particolare? Secondo voi, qual è la direzione verso la quale sta andando il mercato internazionale videoludico in generale? C'è spazio soltanto per le grandi produzioni o il pubblico può ancora essere coinvolto con la creatività ed il mistero forniti da produzioni più piccole o addirittura homebrew?

BDB - Ciò che abbiamo visto accadere negli ultimi anni dimostra senza dubbio che, accanto alle grandi produzioni e ai giochi che puntano su simulazioni della realtà sempre più fedeli e spinte all'estremo tecnologico, possono convivere titoli indie di tutto rispetto, che non disdegnano di strizzare l'occhio al vintage videoludico. A ciò si aggiungono diverse tendenze che certamente permetteranno di creare e diffondere opere più legate all'IF, in alcuni casi anche ad opera di singoli, ma quasi sempre necessariamente da produrre in tandem per ottenere i risultati migliori dall'abbinamento delle rispettive professionalità legate ai contenuti piuttosto che allo sviluppo applicativo.

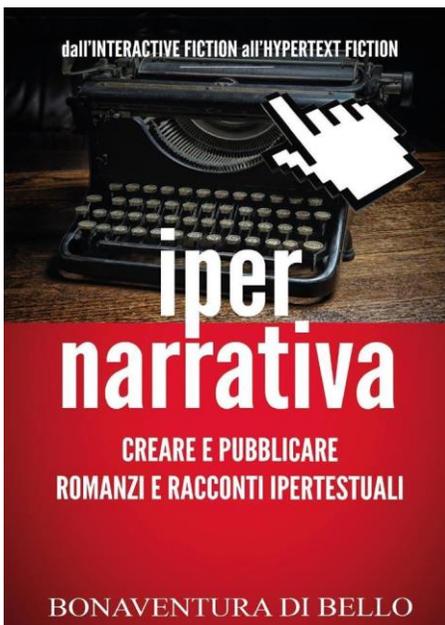
MV - Credo che stiamo andando verso una divisione sempre più netta del mercato e dell'utenza: da una parte ci sono gli hardcore game, prodotti dalle grandi case e rivolti a un pubblico di appassionati che vuole sfide sempre più realistiche, con effetti speciali da kolossal, ore e ore d'esplorazione e spesso un'azione che segue il binario ineluttabile di una trama da film. Dall'altra ci sono i casual game, nei quali secondo me possono rientrare anche le produzioni indie come le IF, che molti cercano in Internet o negli app store per passare il tempo senza troppe pretese e sbattimenti. Quindi, non solo le piccole produzioni continueranno ad avere il proprio spazio ma potrebbero addirittura averne di più, visto che sembra più probabile un incremento del pubblico casual che di quello hardcore. Quello che però non bisogna dimenticare, in fase creativa, è che la gente cerca un divertimento, non una sfida "mortale" che la stressi più del lavoro o della scuola. In un gioco a enigmi, basta poco per rendere una storia intrigante qualcosa di terribilmente difficile o noioso perché non si riesce a andare avanti. Personalmente, da autore, spero di avere sempre gli strumenti adatti a dare vita alle mie idee; da utente, giocatore, lettore, spero di continuare a trovare opere che non mi facciano dimenticare perché ho passato tutti questi anni a giocare, scrivere, leggere, programmare. E sognare.

CONCLUSIONI

Ringraziamo Bonaventura e Marco per la loro ampia e squisita disponibilità e per aver spaziato insieme a noi di RetroMagazine nel passato, nel presente e nel futuro di questa vera e propria arte della scrittura creativa, la fiction interattiva, dove l'immaginazione del giocatore/lettore è, più che negli altri episodi

di intrattenimento elettronico, protagonista assoluta.

Rispetto a tanti anni fa, dove i limiti grafici e sonori delle piattaforme hardware a disposizione in un certo senso contribuivano a far sì che gli autori sviluppassero trame e storie sempre più avvincenti per tenere incollati allo schermo gli utenti dei videogiochi, oggi il mercato sembra scommettere sempre di più (com'è accaduto in altri settori dell'entertainment globale, cinema e musica su tutti) su grandi produzioni e titoli dove l'elemento del consumo immediato e veloce risulta preponderante, purtroppo a scapito della profondità e dello spessore dell'esperienza di gioco. Gli adventure del passato o quelli come Déjà Vu ci mostrano che la creatività e la giocabilità possono essere indipendenti dalla potenza evocativa della tecnologia multimediale e che alcune ore passate a districarsi in un'avventura testuale valgono spesso quanto e forse di più che qualche ora trascorsa in un ambiente 3D iper-realistico a massacrare con armi improbabili alieni o soldati nemici.

È innegabile che a volte un'immagine vale mille parole, ma quelle mille parole, se ben disposte, possono immedesimare e trasportare gli adventurer in un mondo lontano (anche nel tempo oltre che nello spazio) e hanno persino il potere di fare qualcosa di più importante: accendere un pensiero, istigare un'emozione, scatenare nuove idee.

I membri della IF Italia all'Adventure Day 2013 del Vigamus: da sinistra a destra Roberto Grassi, Francesco Cordella, Bonaventura Di Bello, Giovanni Riccardi, Marco Vallarino.

Riferimenti e Bibliografia

[BDB] http://bonaventuradibello.com
[MV] http://marcovallarino.it
Iper-Narrativa - http://bonaventuradibello.com/ipernarrativa
Alla scoperta delle AT (BDB) - http://web.mclink.it/MH4996/
Introduzione alle AT (MV) – http://avventuretestuali.xoom.it/che_cosa_sono_le_avventure_testuali.html

IF Italia - http://ifitalia.oldgamesitalia.net/pmwiki/pmwiki.php
Old Games Italia – http://www.oldgamesitalia.net
Marmellata d'Avventura - http://www.oldgamesitalia.net/content/marmellata-davventura-2018

Scott Adams - https://en.wikipedia.org/wiki/Scott_Adams_(game_designer)
Enrico Colombini - https://it.wikipedia.org/wiki/Enrico_Colombini
The Quill - https://it.wikipedia.org/wiki/The_Quill_(software)
Inform - https://it.wikipedia.org/wiki/Inform

Copia del National Inquisitor distribuito con il gioco Zak McKracken - https://bit.ly/2HCJSA4
Il mitologico uccello Roc presente in Acheton - https://it.wikipedia.org/wiki/Roc

Download di Déjà Vu per C64 (D64] – http://www.retromagazine.net/download/dejavuC64.zip
Seguite le istruzioni nel file LEGGIMI.TXT allegato per eseguire correttamente il gioco.

Gameplay video di Déjà Vu – https://www.youtube.com/watch?v=8KQNp7yzGE8

Progetti: Meteo16

Previsioni Metereologiche con il Commodore 16 di **Daniele Minneci**

Continua la nostra missione di ricerca e divulgazione dei progetti che valorizzano il RetroComputing. Questo mese andiamo a Fucecchio, in Toscana, a conoscere Daniele Minneci, gestore del progetto Meteo16.

Daniele Minneci e' l'amministratore della pagina facebook **MeteoFucecchio.it - Fucecchio Weather Station** seguita da piu' di 3500 persone, dove pubblica regolarmente bollettini meteo riguardanti l'area geografica di Fucecchio.

Come per tanti di noi, anche per Daniele, la passione per il computer nasce negli anni 80 e piu' precisamente nel 1984, quando riceve in regalo un Commodore 16. A quei tempi andava in onda un telefilm che avrebbe segnato l'adolescenza di tanti di noi, 'I ragazzi del computer' ed ovviamente Daniele non ha fatto eccezione. Diversamente da tanti suoi coetanei che vedevano il computer prevalentemente come una macchina da gioco, Daniele si appassiona subito alla programmazione e si interesssa altrettanto alle applicazioni!

Ciao Daniele, da dove nasce la tua passione per la meteorologia?

Sono sempre stato appassionato di Astrologia, ma la passione per la meteorologia nasce per un'esigenza del mio impegno come volontario per la protezione civile. Dopo diversi anni di servizio civile attivo entro a far parte dei centri di situazione regionali che allertano la popolazione in caso di allerte meteo. Da li' mi accorgo che nel territorio regionale di Fucecchio c'e' un gap riguardante le rilevazioni meteo e decido di attivarmi per porre rimedio.

In un primo momento acquisto una stazione meteo amatoriale indipendente (non va in rete), senza connessione pc, a cui poi comincio ad aggiungere componenti, fino alla decisione di acquistare una stazione meteo piu' seria, semiprofessionale...

Per farla breve, adesso posseggo una stazione professionale della **Davis** (fabbricata in America) con sensori esterni interfacciati via wifi con la console interna. Questa stazione, certificata dal produttore, e' riconosciuta a livello internazionale e tramite un Mac Mini, dove gira un programma che gestisce i dati rilevati, li condivide con 7 reti internazionali per poter essere analizzati per previsioni meteo su larga scala sempre piu' attendibili; a livello locale invece, fornisce le sue previsioni.

Quindi la tua stazione meteo fornisce i dati per le previsioni del tempo che vediamo alla televisione?

Purtroppo no. La legislazione italiana in fatto di meteorologia e' piuttosto complicata ed a causa di cavilli burocratici i dati prodotti dalla mia stazione meteo, seppur certificati a livello internazionale, non possono essere utilizzati per le previsioni meteo italiane... Fortunatamente, grazie anche all'impegno dell'amico meteorologo **Gordon Baldacci** che adesso lavora al centro Epson Meteo di Milano, stiamo lavorando per porre rimedio a questa situazione. I vantaggi sono sotto gli occhi di tutti; maggior numero di stazioni meteo, significa piu' dati a disposizione, quindi maggiore affidabilita' nelle previsioni. Inoltre la mia stazione meteo fornisce dati da una zona non coperta dal servizio nazionale...

Torniamo a Meteo 16, puoi raccontarci come hai scoperto il software?

Negli anni 80, in una cassetta contenente utility per il C16 ho scoperto un programma chiamato 'Analisi del tempo', riconosciuto poi come **Meteo 16**, prodotto da **MantraSoft**. Fu amore a prima vista e dal 1985 cominciai ad usare il software in maniera assidua. Creai e stampai dei moduli dove raccogliere i dati da dare in input al software e piano piano cominciai a popolare un file con tutti i dati metereologici del periodo... Conservo ancora quel file che si chiama "MeteoFile".

Come ti e' venuta l'idea di utilizzare il software Meteo 16 per le previsioni meteo attuali?

Sono un appassionato di RetroComputing, possiedo una vasta collezione di macchine Commodore (la linea 264 completa). Il Commodore 16 e' stato il mio primo computer e come tale, come per il primo amore, non si scorda mai. Nel sito Plus4World ho ritrovato il software Meteo16 ed insieme all'amico **Francesco Gori** abbiamo deciso di provare ad utilizzarlo di nuovo per vedere quanto le previsioni meteo elaborate da questo software fossero attendibili. Lo sorpresa e' stata proprio scoprire che lo sono veramente!

Puoi spiegarci brevemente come funziona il calcolo di Meteo 16?

Onestamente non ho ancora avuto modo di analizzare il codice (e' nella mia to do list), qiundi posso solo avanzare delle ipotesi. Tramite i dati forniti in input dall'utente, pressione barometrica, direzione e velocita' del vento, temperatura... il software fa un calcolo basandosi su degli assunti piuttosto semplici: se la pressione barometrica sale il tempo e' piu' stabile, meno perturbato, se invece il barometro scende, significa maltempo... Se scende rapidamente, il maltempo arrivera' velocemente, mentre una salita veloce indica beltempo con probabili condizioni di vento. Sui cambi di temperatura, si basa sulla direzione del vento.

Come inputi i dati su Meteo 16?

Attualmente i dati rilevati dalla stazione Davis vengono inputati manualmente in Meteo16, mentre per il futuro sto pensando di creare un'interfaccia tramite la userport. Il C16 non ha la userport nativa, ma tramite una schedina expansion port potrebbe leggere i dati RS232 della stazione.

Grazie mille Daniele, e' stato un piacere scambiare quattro chiacchiere con te.

Grazie a voi!

E non e' finita qui! Vi anticipo gia' che torneremo di nuovo a parlare con Daniele per scoprire di piu' sull'algoritmo ed i calcoli di Meteo 16!

di Francesco Fiorentini

RetroSpace: Come un pesce in forno: l'incredibile storia di un bug spaziale

di Marco Fanciulli

Questo mese RetroSpace va in ferie ma non vi lascia soli nel vuoto dello spazio!

Riprendiamo qui una storia pubblicata alcuni mesi sul gruppo VCCI a uso e consumo di retroarcheologi in cerca di una lettura prima di andare a dormire.

Lo spazio e l'avvento dei computer digitali

Per ragioni che vi saranno più chiare nei prossimi appuntamenti di RetroSpace, chi vi scrive da molto tempo sta studiando l'architettura del computer di bordo delle missioni Apollo, l'**Apollo Guidance Computer** o **AGC** per gli amici, cercando nelle trascrizioni delle comunicazioni tra il Mission Control Center (MCC) di Houston e gli astronauti delle missioni Gemini e Apollo una conferma rispetto alle sequenze di programmi e comandi eseguiti e impartiti in precisi momenti del piano di volo.

L'AGC era una macchina meravigliosa la quale, per la sua natura peculiare, non trova quasi mai posto nella letteratura del retro-computing, se non tra quella di nicchia della retro-astronautica. È lui il primo eroe di questa storia.

L'AGC Block I è stato uno dei primi computer, se non il primo in assoluto, a fare uso di circuiti integrati e certamente è stato il più compatto della sua epoca, misurando solo 61x32x15cm e pesando "appena" 30kg. Un vero gioiello della tecnologia, studiato per sostituire a bordo di un veicolo spaziale i computer che nei centri di calcolo sulla Terra occupavano intere stanze e avevano bisogno di una centrale elettrica dedicata per poter funzionare.

La straordinarietà dell'AGC è tutta nel riassunto delle sue caratteristiche principali. Si trattava di un computer *realtime* (!), *multitasking* (!!), con capacità interna *di error-recovery* (!!!), costruito attorno ad un bus e una "CPU" a 16 bit (15 di dati e uno di parità), dotato a seconda della release di circa 65-70KB ROM (tra 32k e 38k word da 15 bit) sotto forma di core rope memory (un'invenzione concepita proprio per l'Apollo) e 3.8KB RAM (2048 parole da 15 bit) con una frequenza di accesso e aggiornamento di 85 KHz. Grazie al fatto che ciascun core della memoria "ROM" ospitava 64 fili discreti, i 600.000 bit necessari alla macchina occupavano meno di 10.000 core di memoria, risparmiando parecchio spazio rispetto alle soluzioni al suolo (che nello spazio infinito, paradossalmente, è una cosa buona come lo sarebbe nello spazio finito di casa nostra). Infine, caratteristica di massimo conto, assorbiva solo 2.5A a 28V per 70W di potenza e, per dare una scala dimensionale, aveva una capacità di calcolo paragonabili a quella della triade del '77: il Commodore PET, l'Apple II e il TRS-80 ma quindici anni prima.

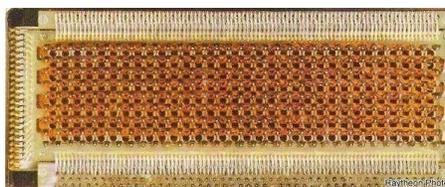



Figura 1: uno stick di core rope memory. 64 fili venivano fatti passare dentro o fuori da un toroide in ferrite a rappresentare gli 0/1 del software che veniva letteralmente "filato" sul telaio

L'AGC era una macchina specializzata per operare nello spazio; di conseguenza la sua architettura era concepita per garantire performance eccellenti nell'esecuzione dei compiti tipicamente legati alla navigazione e al controllo delle complesse equazioni del volo necessarie per determinare la posizione della navicella nello spazio, il suo orientamento, il punto di destinazione e tutte le azioni di spinta e orientamento necessarie a condurvela. In sostanza, era una macchina specializzata nelle operazioni di moltiplicazione e divisione più che rivolta alle operazioni di addizione e sottrazione. Infatti, l'AGC poteva eseguire appena 44.000 addizioni al secondo; pochissime se comparate alle 500.000 di un Commodore PET o alle 384.000 di un coevo DEC PDP-8. Tuttavia l'AGC surclassava entrambi quanto a capacità moltiplicativa, potendo eseguire 22.000 operazioni al secondo contro le sole 300 di un PDP-8 o le 3.000 di un PET, un computer più giovane di oltre un decennio. Questo primato rimase largamente imbattuto fino alle soglie degli anni '80, quando le esigenze del progetto dello Space Shuttle produssero un nuovo step di sviluppo nel settore.

Anche se l'uso della logica integrata era limitata all'adozione di 2.800 NOR gates duali della Fairchild Semiconductor impiegati come flip flop per i registri centrali della CPU, la scelta di limitarne l'uso a questa fattispecie fu presa con estrema ponderazione. Il precedente sistema di guida che aveva sperimentalmente utilizzato una logica diodo-transistor e diodo-diodo (il Minuteman II), aveva dato troppi problemi di affidabilità fin dai collaudi a terra e non ci si fidava molto a portarlo in volo, con le sollecitazioni tipiche di questa condizione. Così gli ingegneri decisero che la CPU dell'AGC non sarebbe stata realizzata in logica IC; senza riaprire l'annoso dibattito in merito a chi abbia concepito e realizzato il primo microprocessore della storia (sappiamo che la logica integrata arriverà solo con il progetto di Ray Holt e l'Intel 4004), bisogna ricordare che alla Grumman non è che non avessero considerato l'idea suggerita dal MIT! Al contrario, avevano valutato attentamente la possibilità di adottare in toto circuiti integrati e di sviluppare una CPU RTL per la quale furono esaminate e vagliate anche le possibili strategie di implementazione (parliamo della stessa Grumman che commissionò poi la medesima CPU alla Garrett presso la quale lavorava Ray Holt, per l'impiego nel futuro F15!). Tuttavia la logica integrata era troppo recente, poco conosciuta e non dava certezze di affidabilità per le condizioni estreme della fase di ascesa, volo e rientro. Per questa stessa ragione tutta la componentistica dell'AGC era **wire wrapped** e poi affogata in un bagno di plastica epossidica, per essere infine chiusa in un contenitore modulare metallico sigillato. Un elemento che, in nome della correttezza storica e a dispetto del costo irrisorio dei singoli componenti, sta facendo

crescere in modo significativo il costo del progetto di cui parleremo nei prossimi articoli.

Figura 2: l'interno dell'AGC evidenzia la sua costruzione modulare e compatta.

Questa meraviglia della tecnologia, un salto tecnologico enorme concepito all'inizio degli anni sessanta e provata in volo appena sei anni dopo, è forse la parte meno appariscente delle missioni Apollo e tuttavia è al centro della missione stessa e, in almeno due casi, della salvezza degli astronauti.

La storia nella storia

Una delle due storie la conosciamo tutti: qualcuno l'ha vissuta in tempo reale nel 1970 mentre a noi altri che eravamo troppo piccoli o nemmeno nati, è stata raccontata da libri, documentari e dal film Apollo 13. L'esplosione di un serbatoio dell'ossigeno a causa di un corto circuito mise a repentaglio la vita dell'equipaggio e solo la fantasia ingegneristica del personale di terra li tenne in vita grazie a calzini, tubi e... alla quadratura del cerchio.

Quella che vi racconto adesso invece è una storia che riguarda l'Apollo 14, in particolare le quattro ore che hanno preceduto fase di discesa del LEM con a bordo Alan Shepard e Ed Mitchell. Mi permetterò alcune inesattezze di minore importanza a favore della linearità della narrazione.

Lo spazio: esterno *giornonotte*

Sono le 10:05 del mattino in Italia ma nello spazio, per gli occupanti del modulo di comando Kitty Hawk che si apprestano a salire a bordo dell'Antares per scendere nell'altopiano di Fra Mauro, è notte. Una notte schizofrenica nel micro mondo della navicella, indecisa tra il buio più nero e la luce accecante riflessa dalla Luna.

Ed Mitchell, che avrebbe pilotato il LEM verso lil complesso atterraggio sull'altopiano, stava effettuando gli ultimi controlli sui computer di bordo dell'Antares mentre Al Shepard - il comandante nonché mito vivente dell'astronautica americana - scorreva la lista dei check con fredda professionalità e rapidità. C'era molto da fare prima di iniziare la gita che tutti al mondo avrebbero voluto (e vorrebbero) fare.

A bordo del LEM, oltre all'AGC, erano presenti altri computer indipendenti e specializzati. Non erano versatili come l'AGC ma avevano piuttosto un unico e immutabile programma prestabilito e essenziale per specifiche fasi del volo. Nel frangente della missione che si svolge durante la nostra storia, in ordine alla sicurezza del volo il più rilevante di questi sistemi funzionali era certamente l'AGS, l'*Abort Guidance System*. Si trattava del computer deputato all'annullamento della fase di atterraggio a seguito di un problema grave quale lo spegnimento del motore principale, la perdita dell'assetto o avarie irreparabili di altra natura.

Ogni 250 millisecondi una routine del'AGC verificava se fosse stato premuto uno dei due pulsanti **ABORT** o **ABORT STAGE**; se quello fosse stato il caso, la routine avrebbe inizializzato un segnalatore discreto che autorizzava l'AGS a prendere il controllo dell'astronave e a dare il via alle manovre di risalita di emergenza attivando il programma **70** (*risalita con motore principale*) o il programma **71** (*risalita da stage intermedi*) ell'AGC.

Scorrendo l'elenco delle verifiche da fare, Mitchell continuava a digitare le sequenze di controllo sul DSKY.

Il DSKY (pronunciato "dischi") costituiva l'interfaccia utente dell'AGC, la prima interfaccia per computer di cui sia documentato l'approccio UX e di ergonomia funzionale; non era insomma un caso che l'AGC fosse un computer semplice da programmare e usare, almeno per gli standard dell'epoca. L'immissione di comandi avveniva attraverso codici numerici che rappresentavano un **VERB** (un'azione o comando) e un **NOUN** (l'oggetto di quel comando), seguiti da eventuali parametri espressi in notazione ottale. L'intero ciclo di volo consisteva in una sequenza di programmi indipendenti, a volte attivati manualmente e a volte attivati automaticamente al verificarsi di determinate condizioni; la stessa attività degli astronauti consisteva quasi integralmente nel richiamare questi programmi o nel passare loro i parametri calcolati a bordo o a terra (i famosi valori elaborati dalle "calcolatrici umane" di cui parla il film "Il diritto di contare" o i dati relativi al cosiddetto "*state vector*" che le cronache riportano come elaborati su una Olivetti P101).

Figura 3: Il DSKY a bordo del LM Antares

Come dicevamo, Mitchell continuava a digitare comandi di impostazione e poi comandi di verifica di queste impostazioni che mostravano sul display a segmenti le relative informazioni di controllo; il tutto mentre la telemetria trasmetteva i risultati al Centro di Controllo affinché potessero essere analizzati e confermati.

Houston, abbiamo un problema anche se non lo sappiamo.

Poco dopo la fase di distacco dal modulo di comando, proprio la telemetria iniziò a segnalare frequenti set e reset dei segnalatori discreti di **ABORT**, come se gli astronauti stessero premendo compulsivamente uno dei bottoni di annullamento di emergenza della discesa.

Ci vollero pochi secondi per individuare il problema in un corto circuito intermittente dovuto probabilmente al distacco di un cavo

durante le fasi di decollo e di inserimento in rotta, fasi piuttosto violente per la struttura di alluminio del complesso Apollo. Ancora una volta un maledetto corto circuito per un cavo rotto stava rischiando di compromettere la missione, a dimostrazione della violenza con la quale la navicella veniva squassata nelle fasi intense del decollo e della assoluta necessità di bloccare tutta la circuiteria con resine e plastiche.

La missione era a rischio e forse anche la vita degli astronauti, in balia di una potenziale accensione indesiderata dei sistemi di risalita se non addirittura del motore principale.

Nonostante il rischio, per il momento il Controllo Missione decise di proseguire ancora per un po' la missione poiché il pericolo non era immediato. Il complesso del sistema di volo aveva infatti una sorta di protezione: un flag di autorizzazione all'annullamento (il flag **LETABORT**) che veniva attivato solo durante l'esecuzione di certi programmi e non di altri (nello specifico il programma **P70** e il programma **P71**). Nel frattempo, sulla Terra, per gli ingegneri erano iniziate le sessioni di *brainstorming* per individuare una possibile soluzione o quantomeno un *workaround* (non essendo possibile una riparazione in volo).

Inizialmente qualcuno pensò di forzare un reset del flag **LETABORT**, così da impedire alle procedure di annullamento di entrare in funzione. Dopo le prime verifiche questa strada non si rivelò applicabile per almeno due ragioni: la prima è che l'accensione del motore di discesa avrebbe comportato l'attivazione automatica del flag **LETABORT** e gli astronauti avrebbero dovuto reimmettere nell'AGC la sequenza di comandi di reset perdendo una decina di secondi durate i quali, se si fosse verificato il cortocircuito, le procedure di risalita sarebbero entrate in funzione con la navicella in assetto ancora orizzontale, probabilmente lanciandola fuori dall'orbita. La seconda è che se si fossa davvero resa necessaria la procedura di annullamento, gli astronauti avrebbero dovuto ripristinare il flag e ripetere l'intera procedura normale di accensione del motore prima di poter passare al programma **P70** (ascesa di emergenza con il motore principale) o **P71** (ascesa in uno degli altri stage del programma di discesa), perdendo secondi preziosi e potenzialmente fatali che avrebbero condotto la navicella a schiantarsi sul suolo lunare.

Mentre queste soluzioni venivano valutate, agli astronauti era stata data solo una sintetica e generica comunicazione in merito a una possibile variazione della procedura di discesa ma non erano ancora stati informati della serietà della situazione; stavano quindi proseguendo le proprie attività previste nel piano di volo senza troppi affanni.

Come un pesce in forno

La Luna non ha un'atmosfera che oppone resistenza a una nave spaziale, così il LEM era stato progettato per il massimo contenimento del peso. Ogni kg di massa risparmiata forniva qualche secondo in più di spinta e controllo al pilota e ogni secondo di controllo in più poteva significare la differenza tra la vita e la morte. Pertanto, alla ricerca della leggerezza estrema, una buona parte della "carrozzeria" del modulo lunare, consisteva in un sottile foglietto di alluminio non troppo diverso da quello con il quale cuociamo il pesce al cartoccio nel forno. Chiusi nel loro piccolo ambiente, separati dallo spazio infinito da una tovaglietta di carta stagnola, Mitchell e Shepard continuavano le loro manovre, programma dopo programma.

Ma, a loro insaputa, il tempo stringeva: rimanevano meno di tre ore per trovare una soluzione ed evitare di dover annullare la missione per non far correre rischi all'equipaggio. La pressione era tantissima.

Dopo i successi dell'Apollo 11 e dell'Apollo 12, l'Apollo 13 aveva fallito e sebbene l'essere riusciti a riportare a casa gli astronauti sani e salvi avesse accresciuto il senso di orgoglio e il supporto alla NASA da parte della popolazione americana, negli ambienti politici si era già messa in dubbio l'utilità di queste missioni a fronte dei costi elevati e in considerazione del fatto che la gara contro i russi era stata ormai vinta. Un secondo fallimento consecutivo avrebbe potuto minare l'intero programma spaziale.

Il workaround

Mentre l'Antares ormai si avvicinava all'inizio della fase di correzione dell'assetto per la discesa, gli ingegneri dell'MIT dovevano ancora elaborare una procedura, comunicarla alla Grumman a Houston per i test sul simulatore e, in caso positivo, poi farla trasmettere dal Controllo Missione agli astronauti per l'inserimento nel "*flight book*".

Circa trenta minuti prima della scadenza del tempo utile entrò in campo, intercettato al bollitore per il caffè, il secondo eroe di questa storia, Don Eyles. Don era uno degli sviluppatori del software dell'AGC del LEM (il software "Luminary"), sotto la guida di Margaret Hamilton. Conoscendo profondamente il funzionamento dell'AGC nella sua versione per il modulo di discesa e dei relativi programmi di volo, trovò immediatamente una possibile via d'uscita nel **MODEREG**, uno dei 13 registri addizionali della CPU dell'AGC.

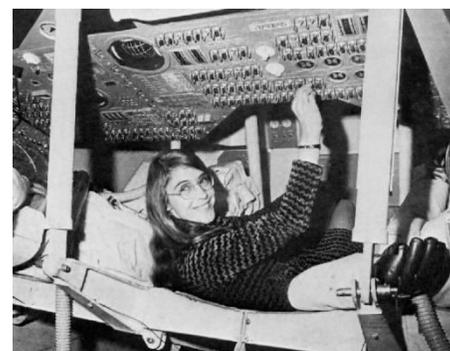



Figura 4: Margaret Hamilton nel simulatore del modulo di comando dell'Apollo 10.

Questo registro indicava la locazione di memoria che conteneva il numero del programma attualmente in esecuzione (il cosiddetto "*major mode*", nel gergo dell'AGC). Il suo scopo era principalmente quello di dare al DSKY l'informazione affinché la mostrasse agli astronauti sul display a segmenti i dati pertinenti a quel programma. Per fortuna quello stesso registro veniva interrogato anche dall'AGS per verificare che i programmi **70** o **71** non fossero già in funzione al momento della richiesta di annullamento della procedura di discesa (in tal caso non avrebbe avuto senso attivare una procedura già in esecuzione!).

Ecco che in pochi minuti il piano prese forma: si doveva ingannare la routine di controllo dell'annullamento inserendo il ***P71*** nel ***MODEREG*** dopo aver manovrato per mettere il LEM in posizione corretta per l'accensione del motore di discesa; si sarebbe poi atteso che il motore, accendendosi, avesse impostato il flag ***LETABORT*** e lo si sarebbe

quindi resettato; se il cortocircuito si fosse verificato nei 10 secondi necessari per la procedura, la routine di controllo dell'annullamento avrebbe trovato P71 nel ***MODEREG*** e non avrebbe lanciato il programma di ***ABORT***. Una volta resettato il flag ***LETABORT***, gli astronauti avrebbero ripristinato il ***MODEREG*** corretto e proseguito la discesa.

Come in un film pieno di suspense, il test presso la Grumman però fallì!

La prova nel simulatore evidenziò come anche la routine di accensione del motore aveva le proprie esigenze! In particolare, si aspettava di trovare nel ***MODEREG*** il Programma ***63*** o non avrebbe portato il motore alla massima potenza dopo i 26 secondi necessari al vettoramento di spinta per la correzione dell'assetto; soprattutto non avrebbe impostato il flag ***ZOOMFLAG*** che segnalava al sistema di discesa di prendere il controllo e portare l'astronave sulla superficie!

Figura 5: il DSKY a bordo del modulo di comando ripreso durante un test a terra prima del lancio.

La soluzione non funzionava ma la strada era segnata. Per circa 20 minuti gli ingegneri lavorarono a una procedura rivista tenendo conto di tutti i vincoli noti; una decina di sviluppatori ripercorrevano i diagrammi di flusso dei programmi coinvolti e le interazioni tra i sistemi con frenesia e alla fine arrivarono alla soluzione: il pilota avrebbe impostato il ***P71*** nel ***MODEREG*** e effettuato le manovre normali fino all'accensione del motore e ai successivi 26 secondi necessari per completare la correzione d'assetto. Quindi il comandante avrebbe portato la manetta al massimo facendo l'*override* del sistema automatico mentre il pilota avrebbe contestualmente impostato lo ***ZOOMFLAG*** per poi resettare il flag ***LETABORT*** e reinserire ***P63*** nel ***MODEREG***. A questo punto Shepard avrebbe riportato la manetta al minimo e il computer di discesa avrebbe preso il controllo. Il vantaggio di questa procedura era che per eseguire una manovra di interruzione di emergenza, l'equipaggio avrebbe dovuto semplicemente "alzare" il flag ***LETABORT*** e premere il pulsante di interruzione appropriato. Pochi istanti di attività a tutto vantaggio della sicurezza.

La flessibilità, programmabilità e usabilità dell'AGC, l'intuizione di un brillante sviluppatore che lo conosceva profondamente e l'addestramento degli astronauti sono il primo esempio di workaround della storia. A 380.000km da qui.

La prossima volta che alzerete gli occhi al cielo e guarderete la Luna, date una sbirciata all'altopiano di Fra Mauro e pensate a quel piccolo grande AGC che è lì, in attesa che un retro-collezionista vada a recuperarlo!

Un po' di storia reale

L'immagine che accompagna in chiusura questo articolo è la trascrizione di quel minuto e ventisette secondi durante i quali tutto sarebbe potuto andare male e invece un piccolo magico computer e un ingegnere che lo amava, salvarono la missione e forse la vita di quegli uomini coraggiosi avvolti in un foglio di carta stagnola.

Nell'estratto rappresentato in figura, la sigla CC sta per Cap Com, il contatto radio a terra che per questa missione era Fred Haise (il pilota del modulo LEM dell'Apollo 13). La sigla LMP-LM sta per Lunar Module Pilot, cioè Ed Mitchell che riuscirà a scendere sulla Luna e a trascorrere complessivamente quasi nove ore passeggiando come mai più nella sua vita. La sigla CDR-LM sta per CommanDeR of Lunar Module e si riferisce a Alan Shepard, un mito assoluto, primo americano nello spazio, il decano degli astronauti a stelle e strisce. Uomo tutto d'un pezzo capace comunque di uscite meravigliose come quella prima del decollo del razzo Redstone che lo avrebbe portato in orbita sulla Freedom 7: "*Please, dear God, don't let me fuck up*". E come dimenticare la sua partita a golf sulla Luna?

Di lui ci rimane anche l'omaggio videoludico nella serie "Mass Effect": è proprio lui il Commander Shepard!

Evidenziata con il bordo rosso è la sequenza di comandi del workaround, numerata per agevolarvi la ricostruzione degli eventi:

1) Dopo che il countdown del NOUN 62 parte (62s che sta per "NOUN 62 starts") Mitchell digita

VERB 21 NOUN 1 ENTER 1010 ENTER 107 ENTER

cioè lancia il comando di ***LOAD*** (***VERB 21***) all'indirizzo ***520*** (***1010*** ottale, l'indirizzo di un registro del display del DSKY che copia il valore del ***MODREG*** - che in realtà ha indirizzo ***500***) del valore ***71*** (***107*** ottale). Cioè mette nel ***MODREG*** letto dalla routine di annullamento l'indicazione che il programma in esecuzione è il programma ***71***.

2-3-4) Dopo 26 secondi esatti dall'accensione del motore principale Shepard porta la manetta a fondo scala (in realtà è di due secondi in ritardo...)

5-6) Mitchell quindi digita :

VERB 25 NOUN 7 ENTER 101 ENTER 200 ENTER 1 ENTER

cioè il comando ***LOAD*** (***VERB 25***, un comando ***LOAD*** diverso dal precedente) una maschera di bit (***NOUN 7***) data dai valori ottali ***101 200 1*** che di fatto attiva le equazioni di discesa per l'atterraggio.

7) Poi digita

VERB 25 NOUN 7 ENTER 105 ENTER 400 ENTER 0 ENTER

che disattiva il monitoraggio dell'annullamento abbassando il flag ***LETABORT*** e alzando il flag ***ZOOMFLAG***

8) Poi digita

VERB 21 NOUN 1 ENTER 1010 ENTER 77 ENTER

per rimettere il ***MODREG*** al suo valore normale ***63*** per permettere al sistema di guida di prendere il controllo.

Tape 70/13
Page 373

Time	Speaker	Text	#
04 11 58 11	CDR-LM	NOUN 62s VERB 21, NOUN 01 ENTER, 10 10, ENTER; 107, ENTER.	1
04 11 58 26	LMP-LM	Okay, Houston. It's in.	
04 11 58 34	CC	Roger, Antares.	
04 11 58 37	CDR-LM	And - Antares is standing by for a PDI GO.	
04 11 58 51	CC	And, Antares; Houston. You're GO for Fra Mauro.	
04 11 58 57	CDR-LM	Good show, Fredo. Thank you.	
04 11 59 00	LMP-LM	Thank you. You troops do a nice job down there - -	
04 11 59 02	CDR-LM	That was beautiful.	
04 11 59 11	LMP-LM	Hey, if you watch us reset, we'll flip the page.	
04 11 59 15	CDR-LM	Let's go.	
04 11 59 16	LMP-LM	Okay.	
04 11 59 25	CDR-LM	...	
04 11 59 27	LMP-LM	Okay. All procedures are normal from here on in except the 26. I actuate the MANUAL THROTTLE to FULL on my side.	
04 11 59 34	CDR-LM	That's correct. I'll start reentering the DPS after you have throttled up.	
04 11 59 39	LMP-LM	Okay.	
04 11 59 42	CDR-LM	Won't have guidance until after I give it to you, after ... Okay.	
04 11 59 52	CDR-LM	We covered everything on that last one?	
04 11 59 55	LMP-LM	Yes, sir.	
04 12 00 00	LMP-LM	At 10 feet per second, we ...	
04 12 00 08	CDR-LM	You're breaking up to me. Would you run your sensitivity up a little?	

Tape 70/14
Page 374

Time	Speaker	Text
04 12 00 10	LMP-LM	Okay. ... sensitivity is full up.
04 12 00 13	CDR-LM	Okay.
04 12 00 25	CDR-LM	Hey, it's a beautiful day in the land of Fra Mauro.
04 12 00 44	LMP-LM	*** will bring MASTER ARM on 30 seconds*** hit us again.
04 12 00 50	CDR-LM	Okay. Houston, the MASTER ARM is ON, and the A and B LIGHTS are ON.
04 12 00 57	CC	Roger, Antares.
04 12 01 19	CDR-LM	Looks quiet; looks good.
04 12 01 28	LMP-LM	MARK; 1 minute.
04 12 01 31	CDR-LM	Okay. And the radar temperature's coming up. Okay.
04 12 01 45	LMP-LM	...
04 12 01 52	MS	...
04 12 01 54	CDR-LM	Okay. The DSKY's on time.
04 12 01 58	LMP-LM	ENGINE ARM to DESCENT.
04 12 02 00	CDR-LM	AVERAGE g is ON. The DESCENT ENGINE is ON.
04 12 02 04	CC	Roger, Antares.
04 12 02 05	CDR-LM	There's ALTITUDE AND VELOCITY light. R_3 looks quiet.
04 12 02 13	LMP-LM	Okay. We're waiting for ULLAGE AUTO light.
04 12 02 18	CDR-LM	R_3 looks good.
04 12 02 21	LMP-LM	ULLAGE.
04 12 02 22	CDR-LM	AUTO ULLAGE.
04 12 02 26	MS	PRO.
04 12 02 27	CDR-LM	3, 2, 1, 0 -

Tape 70/15
Page 375

#	Time	Speaker	Text
2	04 12 02 29	LMP-LM	IGNITION.
	04 12 02 30	CDR-LM	And we have AUTO, IGNITION.
	04 12 02 32	LMP-LM	Ignition looks good.
	04 12 02 33	CC	Roger, Antares.
	04 12 02 35	CDR-LM	We have an AUTO IGNITION.
3	04 12 02 39	LMP-LM	ENGINE ARM, OVERRIDE; ENGINE COMMAND, OVERRIDE.
	04 12 02 41	CDR-LM	Okay. And the MASTER ARM is OFF.
	04 12 02 43	LMP-LM	All right. Standing by for 26.
	04 12 02 45	CDR-LM	Okay. We'll take the THROTTLE up at 26.
4	04 12 02 54	LMP-LM	THROTTLE up.
	04 12 02 56	CDR-LM	Okay. We're at full throttle.
5	04 12 02 58	LMP-LM	The COMMAND is DOWN. VERB 5 - -
	04 12 03 01	CC	Roger, Antares.
6	04 12 03 03	LMP-LM	NOUN 7 VERB 101.
	04 12 03 09	CDR-LM	1.7.
	04 12 03 12	LMP-LM	... have guidance. And you have COMMAND and THROTTLE.
	04 12 03 19	CDR-LM	Okay. We have guidance.
7	04 12 03 23	LMP-LM	All right. I'm DISABLING. VERB 25 NOUN 7 ENTER, 105 ENTER - -
	04 12 03 31	CC	You're GO at 1 Antares.
8	04 12 03 32	LMP-LM	400, ENTER; 0 ENTER. Okay. And landing radar cable, VERB 21 NOUN 01 ENTER; 10 10, ENTER; 77, ENTER. The landing radar is there. Al, you can reduce your THROTTLE to MINIMUM.
	04 12 03 56	CDR-LM	Okay. It's coming down.

Figura 6: trascrizione delle comunicazioni di bordo durante la fase di applicazione del workaround di Eyles

Il resto dei comandi riportati nel punto **8** attiva il radar di atterraggio. Quando tutta la procedura si conclude e il sistema di guida ha preso il controllo, Mitchell chiede a Shepard di portare la manetta al minimo.

"*Okay. It's coming down*" (riferito alla manetta ma è bello pensare che si riferisca al LEM che scende sicuro verso la superficie).

Tutto è bene quel che finisce bene!

Hands on!

Se qualcuno volesse provare a usare l'AGC senza entrare nelle complessità del progetto Virtual AGC, trovate un ottimo simulatore online a questo indirizzo: http://svtsim.com/moonjs/agc.html

Se vi avventurerete in questa prova noterete che dopo aver inserito il comando di clear del flag ***LETABORT*** e di innalzamento di quello ***ZOOMFLAG***, l'indicatore della manetta rimarrà al massimo. Questo avviene, ovviamente, perché rimettendo il ***MODREG*** a 63 il sistema di guida prende il controllo e siamo ancora in una fase del programma di volo nel quale la spinta è massima per ridurre la velocità della navicella. Allora perché Shepard dovette mettere la manetta al minimo se il sistema comunque l'avrebbe tenuta al massimo? Perché la manetta al massimo forzava l'*override* del sistema di controllo e quando questo in seguito avrebbe dovuto parzializzare la spinta, trovando un *override* con manetta al 100% l'avrebbe invece tenuta al massimo con conseguenze orribili.

Buon volo!

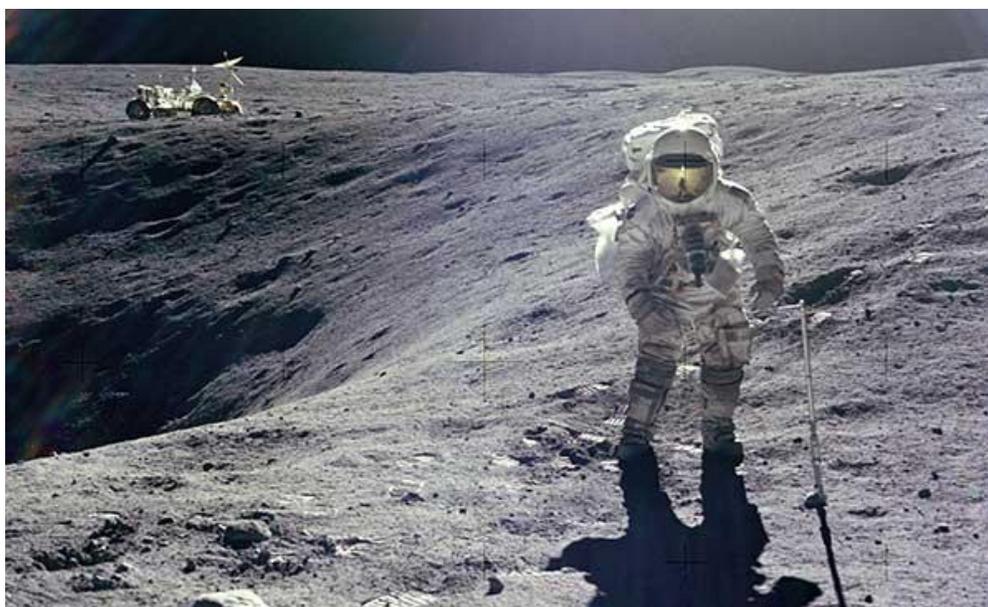



Figura 7: Alan Shepard gioca a Golf sulla Luna.

Chiusura ed anticipazioni...

di Francesco Fiorentini

Universo in espansione...

Secondo la teoria piu' accreditata il nostro Universo si e' generato da una grandissima esplosione (Big Bang) ed e' tuttora in fase di espansione. Espansione tra l'altro dimostrata dallo spostamento verso il rosso delle righe spettrali delle galassie, il cosiddetto red-shift... Ma qui mi sto addentrando in un argomento piu' adatto a RetroSpace che all'articolo di chiusura...

Come per l'Universo, anche la nostra rivista e' ancora in fase di espansione, annoverando ogni mese interessanti novita' (o almeno cosi' crediamo...) da condividere con i lettori.

Come chi ci segue sulla pagina Facebook avra' gia' avuto modo di notare, nel sito internet abbiamo aggiunto la possibilita' di consultare l'**indice degli articoli pubblicati** in RetroMagazine. Navigando all'indirizzo http://www.retromagazine.net/indice_articoli.htm troverete infatti non solo la tabella riportante tutti i pezzi pubblicati sinora nella rivista, ma anche una classificazione degli stessi per rendere piu' semplice e fruibile la loro ricerca ed un link diretto al numero che li contiene per agevolarne una rilettura veloce. Sperando di aver fatto cosa gradita, attendiamo i vostri commenti e suggerimenti per migliorare ancora.

Ma le novita' non si fermano certo qui, altrimenti che realta' in espansione saremmo? Nel prossimo numero o in quello successivo (nel momento in cui scrivo queste righe non mi e' dato ancora saperlo NdR) inaugureremo una rubrica chiamata **RetroGiochiamoli**.

Lo scopo di questa nuova rubrica sara' quello di giocare assieme a voi lettori un gioco, scoprendo cosi' qualcosa di nuovo insieme! Ovviamente per problemi di spazio non potremo giocare tutti i giochi per intero, quindi per lo piu' forniremo soltanto la parte iniziale della soluzione/walkthrough, sperando cosi' di creare abbastanza curiosita' ed interesse affinche' il lettore continui poi da solo. Di soluzioni e walkthrough ne e' piena internet, ma la possibilita' di giocare i giochi condividendone la scoperta e le emozioni provate ci e' sembrata un'ottima idea ed abbiamo voluto provare a farlo sulle pagine della rivista. Anche qui saranno i vostri commenti e reazioni a farci capire se abbiamo colpito nel segno o meno...

Per esigenze editoriali abbiamo dovuto spostare l'ultima parte della guida di Giorgio, **Programmazione dell'Atari 2600** al prossimo numero, ma niente paura, **Giorgio Balestrieri** ci ha promesso un gioco e lo avremo. ☺

Prima di lasciarvi vorrei pero' cogliere l'opportunita' di ringraziare **Gaetano Chiummo** per aver composto e donato a RetroMagazine la musica che accompagna l'intro in assembler di **Marco Pistorio**.

Per il momento non vi anticipo altro, altrimenti scopro troppo le carte...

Al prossimo numero di RetroMagazine!

Disclaimer

RetroMagazine
Anno 2 - Numero 6

Direttore Responsabile
Francesco Fiorentini

Immagine di copertina
Flavio Soldani

Aprile 2018